《붉은별》봉사기용체계
보안봉사대몬관리지도서

차 례

머 리 말

위대한 령도자 **김정일**동지께서는 다음과 같이 지적하시였습니다.

《프로그람을 개발하는데서 기본은 우리 식의 프로그람을 개발하는것입니다. 우리는 우리 식의 프로그람을 개발하는 방향으로 나가야 합니다.》

(《**김정일**선집》 제15권, 196페지)

《붉은별》봉사기용체계 3.0 판에서는 강력한 보안봉사를 제공하고있습니다. 현시기 봉사기관리자들은 봉사에만 노력할것이 아니라 보안의 전초선에 서있다는것을 잊지말아야 합니다. 우리는 보안관리에서 나서고있는 실무적인 자질을 높여 우리의 지식경제산업의 정보를 철저히 보안하여야 합니다.

이 지도서는 봉사기관리자들이 보안관리를 하는데서 실무적으로 나서는 여러가지 보안봉사문제들에 대하여 서술하였습니다.

제 1 절 기동보안

1. 보안평가항목

보안을 평가할 때 다음과 같은것들을 고찰하여야 합니다.

- BIOS 와 BOOT 보안 – 권한없는사용자들이 기계에 물리적으로 접근할수 있는가, 단일사용자기동인가, 암호없이 가능한가?
- 암호보안 – 사용자암호를 어떻게 설정하였는가?
- 관리조종 – 체계관리자가 누구이며 관리조종기능을 얼마나 가지고 있는가?
- 가능한 망봉사들 – 망에서 무슨 봉사들이 있으며 어떤 동작을 하는가?
- 개별적인 방화벽 – 방화벽이 어떤 형태인가, 필요한가?
- 보안강화된 통신도구들 – 어떤 도구들이 통신에 사용되고있는가, 그 보안성능은 어떠한가?

2. BIOS 와 기동적재프로그람에 대한 보안

기동적재프로그람과 BIOS(혹은 BIOS 와 동일한) 를 위한 통과암호는 분리가능한 매체를 사용하거나 혹은 단일사용자방식을 통하여 root 권한을 얻는 물리적으로 접근가능한 권한없는 사용자들을 막을수 있습니다.

1) BIOS 통과암호

BIOS 통과암호를 걸어야 하는 2 가지 중요한 리유가 있습니다.

1. BIOS 설정의 변환을 막기 위하여 필요합니다. – 공격자가 BOIS 에 접근하여 기동설정을 CD-ROM 이나 다른것으로 설정할수 있습니다. 이것은 체계를 safe 방식 혹은 단일사용자방식에 들어가게 할수 있는데 이렇게 되면 자료들에 대한 자유로운 처리를 진행할수 있게 됩니다.
2. 체계기동을 막기 위하여 필요합니다. – 일부 BOIS 들은 기동할 때 통과암호를 요구합니다. 이렇게 되면 공격자가 안내적재프로그람을 기동하기전에 통과암호를 입력하여야 함으로 한단계의 보안을 강화할수 있습니다.

BOIS 통과암호를 설정하는 방법들은 콤퓨터제작회사들에 따라 다르기

때문에 그에 해당한 사용설명서들을 보아야 합니다.
　만일 이 암호를 잊어버리는 경우 CMOS 축전지를 분리하여 재설정할 수 있습니다.

2) 기동적재프로그람통과암호

다음과 같은 우점이 있습니다.

1. 단일사용자방식에로의 접근을 막습니다. – 만일 공격자들이 단일사용자방식으로 체계를 기동한다면 root 암호를 모르고도 root 의 권한을 가지게 됩니다.

2. GRUB 조작탁 에로의 접근을 막습니다. – 만일 기동적재프로그람으로 GRUB 를 사용하고있다면 GRUB 편집대면부를 사용하여 cat 지령으로 정보를 수집하거나 설정을 바꿀수도 있습니다.

3. 보안기능이 없는 조작체계에로의 접근을 막습니다. – 만일 두개의 기동체계가 있다면 공격자는 기동할 때 조작체계를 설정할수 있습니다.(실례를 들어 DOS) 그러면 마음대로 조종을 진행하게 됩니다.

GRUB 보안 암호

만일 기동적재프로그람통과암호에서 첫 두가지문제의 해결을 위하여 GRUB 를 설정할수 있습니다. 이것을 설정하기 위하여 우선 강한 암호를 선택한 다음 root 로 로그인하여 쉘창문을 열고 다음과 같은 지령을 실행합니다.

/sbin/grub-md5-crypt

다음 GRUB 암호를 입력하고 Enter 건을 누릅니다. 그러면 MD5 압축으로 암호가 채워집니다.

다음 GRUB 설정화일(/boot/grub/grub.conf)을 다음과 같이 편집합니다.

Password –md5 <password-hash> 에서 <password-hash>를 /sbin/grub-md5-crypt 로 설정합니다.

그러면 재기동할 때 GRUB 암호가 없이는 편집기나 지령창문에 접근할수 없게 됩니다.

그러나 두개의 기동환경에 있는 보안기능이 없는 조작체계에로의 공격자들의 공격은 막지 못합니다. 이를 위하여 /boot/grub/grub.conf 화일을 편집해주어야 합니다.

보안하려는 조작체계의 title 행을 보고 그 lock 행을 추가하여야 합니다.

실례로 DOS 체계에 대하여서는 다음와 같이 설정합니다.

title DOS lock

체계에 통과암호를 설정하려면 다음과 같은 지령을 실행하면 됩니다.

title DOS lock password --md5 <*password-hash*>

3. 통과암호보안

《붉은별》봉사기용체계3.0판에서 통과암호는 사용자식별자를 검증하기 위하여 리용하는 초보적인 방법입니다. 이러한 리유로 사용자나 본체, 망을 보호하기 위하여 통과암호보안은 매우 중요합니다.

보안목적으로 MD5와 그림자암호(shadow password)를 사용한 체계를 설치하는데 이것은 매우 중요합니다. 만약에 설치과정에 MD5암호를 설정하지 않는다면 기정의 Data Encryption Standard(DES)방식이 사용되게 됩니다. 이 방식으로는 최대길이 8글자의 통과암호 즉 최대로 56비트준위의 통과암호밖에 설정할수 없습니다.

만일 그림자암호를 선택하지 않으면 모든 통과암호들이 /etc/passwd화일에 한방향하쉬로 보관되게 됩니다. 이것은 공격자들에게 통과암호를 공개하는것이나 다름이 없습니다. 그러므로 이러한 보안기능이 없는 통과암호들이 설정되여있다면 암호공격자들이 그것을 발견하기전에 무조건 퇴치하여야 합니다.

그림자암호는 /etc/shadow화일에 하쉬되여 보관되기때문에 오직 root사용자만이 그것을 열수 있습니다.

공격자들은 SSH나 FTP와 같은 망봉사로 원격으로 들어갈 때 통과암호공격을 시도합니다. Brute-force공격은 매우 느리며 또 체계화일들에 수백번의 암호공격을 진행하면 폭주상태로 될수 있습니다. 이때 공격한 리력이 남게 되는데 물론 공격자들은 이와 같은 리력을 삭제하거나 편집합니다. 그러나 만약 약한 암호를 설정하였다면 이와 같은 공격이 가능해지게 됩니다. 그러므로 통과암호는 공격자들의 암호공격을 막기 위하여 강한 암호로 설정하여야 합니다.

1) 강력한 통과암호생성

통과암호를 설정할 때 다음과 같은 원칙들을 지키면 강력한 통과암호로 될수 있습니다.

● **문자나 수자만으로 이루어진 암호를 사용하지 말아야 합니다.** – 암호가 문자나 수자만으로 이루어지면 안됩니다. 실례로

- 48382923
- Juan
- hackme

● **잘 알려진 단어들을 사용하지 말아야 합니다.** – 이름이나 사전단어, TV나 책에서 나오는 단어들은 피해야 합니다. 만일 그뒤에 수자가 있다고 하여도 마찬가지입니다. 실례로

- john1
- DS-9
- Mentat132

● **외국어단어들은 사용하지 말아야 합니다.** – 암호공격프로그람들은 많은 언어를 가진 압축된 사전을 사용합니다. 통과암호로 외국어단어를 리용한다면 그것은 안전하지 못합니다. 실례로

- cheguevara
- bienvenido1
- 1dumbKopf

● **해커전문용어를 사용하지 말아야 합니다.** – 해커전문용어(1337 혹은 LEET 언어)를 통과암호로 설정하면 안전하다고 생각할수 있는데 대부분의 단어표들은 LEET를 포함하고있다는것을 알아야 합니다. 실례로

- H4X0R

- 1337

● **개인정보를 사용하지 말아야 합니다.** – 만일 공격자가 사용자의 특성을 알고있다면 암호를 추측하는것은 훨씬 쉬워질것입니다. 통과암호를 설정할 때 피해야 할 항목들의 실례는 다음과 같습니다.

- 사용자의 이름
- 애완용동물의 이름
- 가족사람들중의 이름
- 생년월일
- 전화번호나 생략코드

● **잘 알려진 단어들을 거꾸로 한 통과암호를 사용하지 말아야 합니다.** – 암호를 검사하는 사람들은 항상 일반적인 단어들을 거꾸로 생각하군 합니다. 그러므로 이와 같은 암호들도 좋지 못한 암호로 됩니다.

- R0X4H
- Nauj
- 9-DS

● **통과암호를 적어놓지 말아야 합니다.** – 암호를 종이에 적어 놓아서는 안됩니다. 그보다 머리속에 기억하고있는것이 훨씬 더 안전합니다.

● **모든 기재들에 같은 암호를 사용하지 말아야 합니다.** – 매 기재마다 서로 다른 암호를 설정하는것은 매우 중요합니다. 이렇게 하면 하나의 체계가 공격당해도 다른 체계들이 인차 위험에 처해지지 않을수 있기때문입니다.

강력한 통과암호를 설정하는데 도움을 줄수 있는 몇가지 항목들을 서술하였습니다.

● 통과암호의 길이를 최대한 8글자이상으로 설정하여야 합니다. – 통과암호가 길수록 좋습니다. 만일 MD5암호를 리용한다면 15글자이상 되여야 합니다. DES암호들은 최대길이가 8글자입니다.

● 큰 글자와 작은 글자를 섞어서 설정하여야 합니다.

● 글자와 수자들을 섞어서 설정하여야 합니다. – 특히 통과암호의 시작이나 끝이 아닌 가운데 섞어놓으면 더 좋습니다.

● 특수기호들을 포함하여야 합니다. - &, $, >와 같은 특수기호들은 통과암호의 성능을 향상시키게 됩니다(이것은 DES암호를 사용할 때에는 불가능합니다).

● 사용자가 기억할수 있는 암호를 선택하여야 합니다. – 만일 사용자가 기억할수 없는것이라면 제일 좋은 통과암호라고 하여도 그리 좋다고 할수 없습니다.

통과암호설정방법

통과암호를 설정하는 많은 방법들이 있습니다.

● 기억하기 쉬운 문구 "over the river and through the woods, to grand mother's house we go."

● 구두점을 포함한 준말(즉 기억하기 쉬운 문구의 매 단어 앞글자만 따서 만든 통과암호) – otrattw,tghwg.

● 기억하기 쉬운 준말들가운데서 일부 문자를 수자나 특수기호로 바꾸어 기억하는 방법 – o7r@77w,7ghwg.

● 적어도 한글자는 대문자로 바꾸는것 – 즉 o7r@77w,7gHwg.

● 다른 체계들에서는 이 암호를 쓰지 않는것

2) 사용자그룹에서 통과암호 설정

많은 사용자들을 가지고있는 사용자그룹에서는 체계관리자들이 좋은 통과암호를 사용하기 위한 두가지 기초적인 방안이 있습니다. 관리자는 사용자를 위한 통과암호를 생성할수 있으며 혹은 사용자자신이 만든 통과암호를 허락할수 있습니다. (물론 그 통과암호의 질이 담보될 때)

사용자들을 위하여 만든 통과암호는 그 암호가 좋다는것을 담보하지만, 사용자의 수가 계속 불어나게 되면 그것도 곤란하게 됩니다. 이러한 리유로 대부분의 체계관리자들은 사용자 자신들이 암호를 만들게 하고 그를 검증하여 좋다는것을 담보하며 또한 주기적으로 암호를 교체하게 합니다.

암호갱신

암호갱신은 체계관리자들이 그룹안에서 나쁜 암호를 없애기 위하여 쓰는 또 하나의 방법입니다. 즉 암호갱신의 의미는 특정한 기간 (보통 90일)이 지나면 사용자들이 새로운 암호를 설정할것을 요구합니다. 암호갱신의 필요성은 만일 암호가 해득되였다고 하여도 그 암호를 제한된 시간만큼만 리용하게 되기때문입니다.

《붉은별》봉사기용체계3.0판에는 2개의 기본적인 암호갱신프로그람이 있습니다. 그것은 chage명령과 User Manager(system-config-users)프로그람입니다.

Chage 명령에서 설정값 -M 은 암호가 유효한 기간을 설정하는 값입니다. 실례로 기간을 90일로 설정하려면 "chage –M <username>" 과 같이 지령을 실행시키면 됩니다.

우의 지령에서 이 설정값의 최대값은 99 999입니다. 이것은 273년에 해당됩니다.

구체적인 설정을 위하여서는 "chage <username>"지령을 실행시키면 됩니다.

[root$myserver ~]# change kim
Changing the aging information for kim
Enter the new value, or peess Enter for the default
Minimum password Age [0] : 90
Maximum password Age [99999] : 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) :

Chage 에 대한 도움말을 보면 자세한 설정값들에 대하여 파악할수 있습니다.

3) 봉사들에 대한 위험성

봉사기용체계에서의 망보사들은 많은 위험성들을 가지고있습니다. 대표적인 내용들의 일부를 보여줍니다.

- 봉사거부공격(DoS)
- 분산형봉사거부공격(DDos)
- 스크립트취약성공격
- 완충기넘침공격

주의: 망에서 공격을 제한하기 위하여 사용하지 않는 모든 봉사들을 정지하여야 합니다.

4) 식별자와 해당된 봉사들

최대의 보안을 위하여 《붉은별》봉사기용체계3.0판을 설치하고 모든 망봉사들을 기정으로 정지시키고있습니다. 그러나 다음과 같은것은 제외입니다.

- cupsd – 《붉은별》봉사기용체계3.0판의 기정인쇄봉사기
- xinetd – gssftp 와 telnet와 같은 부분봉사기들에 대한 접속을 조종하는 최고봉사기
- sendmail – Sendmail Mail transport Agent(MTA)는 기정으로 능동설정되여있습니다. 그러나 localhost로부터의 접속을 위하여서만 동작합니다.
- sshd – OpenSSH 봉사기는 Telnet를 위한 보안갱신입니다.
- Beam-이것은 관리할때만 있고 중지하여야 합니다.
- Mysql-자료기지도 중지시켜야 합니다.

5) 보안기능이 없는 봉사들

완벽하게 보안된 봉사는 없습니다. 이러한 리유로 중요하지 않거나 리용하지 않는 봉사들을 끄는것이 매우 중요합니다. 또한 봉사들에 대한 공격을 막기 위하여 패치를 진행하고 망봉사와 관련한 패키지들을 갱신하는것이 매우 중요합니다.

일부 망통신들은 특별한 보안기능이 없습니다. 다음과 같은 봉사들을 포함하는 통신들을 보면.

- 망으로 암호화되지않은 사용자이름과 통과암호를 전송하는것 – Telnet와 FTP와 같은 많은 오래된 통신들은 인증쎄션을 암호화하지 않습니다.
- 망으로 암호화되지않은 가능한 자료의 전송 – 망에서 암호화되지 않은 자료통신규약들이 많이 있습니다. 여기서는 Telnet, FTP, HTTP, SMTP와 같은 많은 것들이 포함되여있습니다. NFS와 SMB와 같은 많은 망화일체계들도 망에서 자료전송을 암호화하지 않고 진행합니다.

Netdump 와 같은 Remote memory dump 봉사들도 망에서 암호화가 없

이 자료전송을 진행합니다.
Finger나 rwhod와 같은 다른 봉사들은 체계의 사용자들에 대한 정보를 나타냅니다.

일반적인 보안기능이 없는 봉사들은 rlogin과 rsh, telnet, vsftpd입니다.

모든 원격가입과 셸프로그람(rlogin, rsh, telnet)들은 SSH에 의하여 철저히 해결되여있습니다.
FTP는 체계보안이 진행되여있으므로 그렇게 위험한것으로 되지는 않지만 그에 대한 깊은 주의를 돌려야 하며 일련의 문제들을 해결하여야 합니다.
주의를 돌려야 할 봉사들은 다음과 같습니다.

- finger
- authd
- netdump
- netdump-server
- nfs
- rshod
- sendmail
- smb(Samba)
- yppasswdd
- ypserv
- ypxfrd

4. 개별적인 방화벽들

필요한 망봉사들을 기동시킨후에 방화벽을 설치하는것이 중요합니다.
주의: 인터네트나 잘 믿음이 안가는 망으로 접속하기 전에 필요한 봉사를 기동시키고 방화벽을 무조건 설치하여야 합니다.

방화벽은 체계의 망대면부에 접근하는 망케트들을 막습니다. 만일 어떠한 요구가 방화벽에 의하여 닫겨진 포구로 들어오게 되면 그 요구는 무시됩니다. 또한 이러한 닫겨진 포구들로 진행되는 봉사는 파케트들을

접수하지 않으며 효과적으로 비활성화됩니다. 이러한 리유로 사용하지 않는 포구들의 접근을 막기 위하여 방화벽을 쓰는것이 매우 중요하게 됩니다.

대부분의 사용자들은 《붉은별》봉사기용체계3.0판에서 iptables를 사용합니다.

5. 보안강화한 통신도구들

인터네트의 규모와 인기가 점점 증가할수록 통중요의 자료를 가로채는 징후도 점점 많이 나타나고있습니다. 여러해에 걸쳐 자료들이 망을 통하여 전달되기때문에 통신에서 기호화를 위한 도구들이 많이 개발되였습니다.

《붉은별》봉사기용체계3.0판에서는 망에서의 자료전송의 보안을 위하여 high-level과 public-key-cryptography에 기초한 암호화 알고리듬을 리용한 2개의 기본적인 도구들을 주었습니다.

- OpenSSH – 망통신의 암호화를 위한 SSH규약의 자유로운 실현
- Gnu Privacy Guard(GPG) – 자료암호화를 위한 PGP암호화응용프로그람의 자유로운 실현

OpenSSH는 원격기계를 접속하거나 telnet와 rsh와 같은 비암호화된 봉사들을 교체하는데서 더 안전한 방법입니다. OpenSSH는 sshd라고 하는 망봉사를 포함하며 3개의 지령행을 가집니다.

- ssh – 원격조작탁접근의뢰기보안
- scp – 원격복사지령보안
- sftp – 대화식화일전송쎄션을 허용하는 pseudo-ftp의뢰기 보안

GPG는 개인용전자우편통신을 담보하기위한 한가지 방법입니다. 공유된 망을 통하여 중요한 자료를 전자우편으로 보내고 또 하드구동기에 있는 중요한 자료를 보호하는것은 둘다 필요한 가능성이 있습니다.

제 2 절 봉사기보안

망에서 체계가 봉사기로 사용될 때 그것은 공격의 목표로 될수 있습니다. 체계를 강화하고 봉사들을 페쇄하는것은 체계관리자가 할 중요한 사업입니다.

구체적인 내용을 고찰하기 전에 먼저 다음과 같은 일반적인 사항들을 고려하여야 합니다.

- 최근공격을 막기 위한 현재의 모든 봉사들을 유지하여야 합니다.
- 가능한껏 보안규약을 사용하여야 합니다.
- 가능하면 한대의 콤퓨터에 하나의 망봉사만을 진행하여야 합니다.
- 봉사기들의 모든 활동들을 주의깊게 고찰하여야 합니다.

1. TCP Wrapper 와 xinetd 에 대한 보안봉사들

TCP Wrapper는 여러가지 봉사들에 접근조종기능을 제공합니다. SSH, Telnet, FTP와 같은 대부분의 현대적인 망봉사들은 TCP Wrapper를 사용하는데 이것은 들어오는 요구와 요구한 봉사들사이를 방어하는 기능을 수행합니다.

TCP Wrapper에 의한 우월성은 추가적인 접속, 호출, 결합, 재지시, 자원리용조종을 진행하는 최고봉사기 xinetd와 결합하여 사용될 때 더욱 더 나타나게 됩니다.

1) TCP Wrapper 를 가지고 보안을 강화

TCP Wrapper는 봉사들에 대한 호출을 거절하는것보다 훨씬 더 유용합니다. 이 절에서는 그것들이 어떻게 련결기발을 보내고 특별한 호스트로부터의 공격에 대하여 경고하며 접속기능을 강화하는데 사용할수 있는가를 보여줍니다.

TCP Wrapper 와 련결기발(Connection Banners)

적합한 기발을 표시하는것은 사용자들이 봉사에 련결될 때 체계관리자가 경계하는 잠재적인 공격자들을 알수 있게 하는 좋은 방법입니다.

또한 체계에 대한 어떠한 정보를 사용자들에게 제공하겠는가를 조종할 수 있습니다. 봉사를 위한 TCP Wrapper Banner 를 실현하기 위하여 banner option을 사용하여야 합니다.

우선 체계의 임의의 곳에 banner 화일을 생성합니다. 그러나 이때 대몬과 같은 이름으로 하여야 합니다. 실례로 /etc/banners/vsftpd를 들수 있습니다.

220-Hello, %c

220-All activity on ftp.example.com is logged.

220-Inappropriate use will result in your access privileges being removed.

여기서 %c 는 사용자이름과 호스트이름 혹은 사용자이름과 IP 주소와 같은 여러가지 의뢰기정보입니다.

여기서 기발은 /etc/hosts.allow화일에 추가됩니다.

vsftpd : ALL : banners /etc/banners/

TCP Wrapper 와 공격경고(Attack Warnings)

만일 어떤 호스트나 망이 봉사기공격으로 검출되였으면 TCP Wrapper 는 spawn 을 사용하여 그 호스트나 망으로부터 련속적인공격의 관리자를 경고하는데 사용될수 있습니다.

실례로 206.182.68.0/24망에서 크랙커가 봉사기를 공격하려고 하다가 검출되였다고 가정해보자. 이때 /etc/hosts.deny에 행을 추가하여 그 망으로부터 임의의 련결시도들을 거절하고 특수화일의 시도를 기록하게 됩니다.

TCP Wrapper 와 강화된 가입

련결의 정확한 형태에 대하여 관심을 높이기 위하여서는 접속준위를 severity option을 리용하여 봉사를 강할수 있습니다. 실례를 들어 누군가가 FTP봉사기에 23번포구(telnet 포구)를 리용하여 접속하려 한다면 그것은 크랙커라고 가정하면 됩니다. 이를 위하여 표준기발대신에 기록화일들에 emerg기발을 배치하고 그러한 련결을 거절하면 됩니다.

이를 위하여서는 /etc/hosts.deny에 다음과 같은 지령을 실행하면 됩니다.

in.telnetd : ALL : severity emerg

2) xinetd 에 의한 보안강화

xinetd를 리용하여 봉사의 방책을 설정하고 또 어떤 주어진 xinetd봉사를 위한 리용가능한 자원준위를 조종하는데 기본초점을 둡니다. 봉사들의 자원한계를 설정하는것은 DoS공격을 막는데 도움을 줄수 있습니다. xinetd와 xinetd.conf를 위한 도움말을 보면 가능한 설정값들을 알수 있습니다.

방책설정

xinetd의 중요한 사명의 하나는 전체적인 no_access표에 host들을 추가하는 능력입니다. 이표에 있는 host들은 정해진 시간동안 혹은 xinetd가 재기동할때까지 xinetd에 의하여 관리되는 봉사들에로의 부분접근을 할수 없습니다. 이를 위하여 SENSOR속성을 리용할수 있습니다. 이것은 봉사기의 포구검사를 진행하여 host를 관리하기위한 쉬운방법중의 하나입니다.

SENSOR를 설정하기 위한 첫 단계는 사용하지 않기로 결정한 봉사를 설정하는것입니다. 다음과 같은 실례에서는 telnet를 설정하였을 때를 보여주었습니다.

/etc/xinetd.d/telnet화일을 편집하고 Flags 를 변환합니다.
Flags = SENSOR
다음행에 다음과 같이 추가합니다.
dny_time = 30

이렇게 하면 30분동안 host에의하여 포구에 접속하는 모든 접속을 거부합니다. 여기서 deny_time속성을 FOREVER로 설정할수도 있습니다. 이렇게 하면 xinetd를 재기동할까지 접속을 거부하며 NEVER로 설정하면 접속을 허용하고 그것을 기록합니다.

마지막에 다음과 같은 행을 추가합니다.
dsable = no

SENSOR를 리용하는것은 비정상적인 호스트로부터의 접속을 검출하고 저지시키는 좋은 방법중의 하나이지만 이것은 2가지 약점을 가집니

다.

● stealth 검사에 대응하지 못합니다.

● 만일 SENSOR가 돌아가고있는것을 알고있는 공격자가 있다면 IP 주소를 속이고 금지된 포구로 접속하여 부분적인 호스트들에 DoS공격을 할수 있습니다.

봉사기자원조종

xinetd의 다른 또 하나의 중요한 점은 그것이 조종하는 봉사들의 자원 한계를 설정하는 능력입니다.

그것은 다음과 같이 진행합니다.

● cps = <number_of_connections> <wait_period> - 들어오는 접속의 시간제한을 줍니다.

<number_of_connections> - 초당 조종할수 있는 접속개수입니다. 만일 들어오는 접속의 개수가 이보다 높으면 봉사는 실행되지 않습니다. 이것의 기정값은 50입니다.

<wait_period> - 봉사가 비활성화 되였다가 다시 기동할 때까지 기다림 시간입니다. 이것의 기정값은10초로 되여있습니다.

● instances = <number_of_connections> - 봉사가 허용하는 접속의 총 개수를 지정합니다. 이것은 옹근수값이나 UNLIMITED로 설정될수 있습니다.

● per_source = <number_of_connections> - 매 호스트에 대한 허용되는 봉사의 개수를 지정합니다. 이것은 옹근수값이나 UNLIMITED로 설정될수 있습니다.

● rlimit_as = <number [K|M]> - 봉사가 차지하는 기억주소공간의 크기를 키로바이트 혹은 메가바이트단위로 지정합니다. 이것도 옹근수값이나 UNLIMITED로 설정될수 있습니다.

● rlimit_cpu = <number_of_seconds> - 봉사가 CPU를 차지할수 있는 시간을 초단위로 지정합니다. 이것도 역시 옹근수값이나 UNLIMITED로 설정될수 있습니다.

이와 같은 설정값들을 리용하면 DoS공격과 같은 공격들을 막는데 도움을 줄수 있습니다.

2. Portmap 보안

Portmap봉사는 NIS나 NFS와 같은 RPC봉사들을 위한 동적포구할당대몬입니다.

그것은 약한 인증기능을 가지고있으며 그것이 조종하는 봉사들에 대한 넓은 포구범위를 설정하는 능력을 가집니다.

주의 : portmap 보안은 오직 NFSv2과 NFSv3의 실현에서만 효과가 있습니다. NFSv4는 그것을 요구하지 않습니다. 그러므로 만일 NFSv2이나 NFSv3을 실현하려고 한다면 Portmap가 요구될것입니다.

1) TCP Wrapper에 의한 portmap 보안

이것은 그 어떤 인증형식도 가지지 않기때문에 망이나 호스트들의 portmap봉사에로의 접근을 제한하는데서 TCP Wrapper를 리용하는것은 매우 중요합니다.

이때 봉사에로의 접근을 제한하는데서 IP주소들만을 사용하여야 합니다. DNS사용이나 다른 방법들에 습관되여있을수 있으므로 hostname의 사용을 피해야 합니다.

2) iptables 에 의한 portmap 보안

portmap봉사에로의 접근제한을 위하여 봉사기에 iptables규칙들을 추가하고 특정한 망에로의 접근을 제한하는것이 좋습니다.

2개의 iptables지령실례가 있습니다. 첫번째것은 192.168.0.0/24망으로부터 포구111(portmap봉사에 의하여 리용된)에로의 TCP련결을 허용하는 실례이며 두번째것은 localhost로부터 같은포구에로의 TCP련결을 허용하는 실례입니다. 이것은 Nautilus에 의하여 사용된 sgi_fam봉사를 위하여 필요합니다.

```
iptables –A INPUT –p tcp –s! 192.168.0.0/24 –dport 111 –j DROP
iptables –A INPUT –p tcp –s 127.0.0.1 –dport 111 –j ACCEPT
```

UDP전송제한을 간단히 다음과 같은 지령을 리용하여 진행할수 있습니다.

```
iptables –A INPUT –p udp –s! 192.168.0.0/24 –dport 111 –j DROP
```

3. Apache HTTP 봉사기보안

Apache HTTP봉사기는 《붉은별》봉사기용체계3.0판에 내장되여있는 가장 안전하고 보안이 좋은 봉사들중의 하나입니다. 많은 설정들과 기술들이 Apache HTTP 봉사기를 보안하는데 쓰이는데 그중 중요한 몇가지 규칙들만을 소개하려고 합니다.

항상 체계에서 동작하는 모든 스크립트들이 제품에 들어가기전에 정확히 동작하는가를 검사하여야 합니다. 또한 root사용자만이 스크립트와 CGI들을 포함하는 임의의 등록부에 쓰기권한을 가진다는것을 확정하여야 합니다. 이를 위하여 root사용자로서 다음과 같은 지령을 실행시켜보면 됩니다.

Chown root <directory_name>

Chmod 755 <directory_name>

체계관리자들은 다음과 같은 설정값들을 설정할 때 주의를 돌려야 합니다. (/etc/httpd/conf/httpd.conf 에서 설정합니다.)

FollowSymLinks

이 지령은 기정으로 제공되여있습니다. 그러므로 웨브봉사기의 root문서들에서 기호련결을 생성할 때 주의를 돌려서 사용하여야 합니다. 실례로 련결기호를 /로 하는것은 좋지 못한 생각입니다.

Indexes

이 지령은 기정으로 제공되여있습니다. 그러나 이것은 불필요한것일수도 있습니다. 봉사기에 화일열람을 위해 접속하는 사람들을 막기 위하여 이 명령을 제거하여야 합니다.

UserDir

이 지령은 기정으로 비활성화되여있습니다. 왜냐면 체계에서 사용자식별이름의 존재를 확증할수 있기때문입니다. 망에서 사용자등록부열람을 위하여 다음과 같은 지령을 사용하여야 합니다.

userDir enabled

userDir disabled root

4. FTP 보안

화일전송통신규약(FTP)는 망에서 화일전송을 위하여 설계된 오래된 TCP통신규약입니다. 사용자인증을 포함하는 봉사기의 모든 트랜잭션(업무)들이 모두 비암호화되여있기때문에 그것은 보안기능이 없는 전송규약으로 고찰되며 여기에 깊은 주의를 돌려야 합니다.

《붉은별》봉사기용체계3.0판은 vsftpd를 사용합니다.

지도서는 vsftpd에 의한 FTP봉사의 설정을 서술하였습니다.

1) FTP Greeting Banner

사용자이름과 통과암호를 접속하기전에 모든 사용자들은 greeting banner를 받습니다. 기정으로 판본정보를 가지고있는 이 기발은 크랙커들이 체계에서 취약점을 발견하는데 필요합니다.

vsftpd를 위한 greet기발을 변화하기 위하여서는 다음과 같이 /etc/vsftpd/vsftpd.conf 화일을 보면 됩니다.

ftpd_banner=<insert_greeting_here>

여기서 <insert_greeting_here>는 greeting message의 본문을 의미합니다.

복합기발을 위하여서는 기발화일을 사용하는것이 제일 좋습니다. 복합기발의 관리를 진행하기 위하여 /etc/banners/ 라는 새로운 등록부를 만들고 여기에 모든 기발들을 배치합니다. 이 례에서는 FTP련결을 위한 기발화일을 /etc/banners/ftp.msg 로 하였습니다. 서 해당한 실례를 보여주었습니다.

vsftpd를 위한 인사기발화일을 참조하기 위하여서는 /etc/vsftpd/vsftpd.conf 화일을 다음과 같이 편집하면 됩니다.

banner_file=/etc/banners/ftp.msg

2) 닉명접속(Anonymous Access)

/var/ftp/ 등록부의 존재는 닉명의 사용식별자를 찾아내는데 도움을 줍니다. 이 등록부를 생성하는 가장 쉬운 방법은 vsftpd패키지를 설치하는 것입니다. 이 패키지는 닉명의 사용자들에 대한 등록부나무를 가지고있으며 이러한 사용자들에게 읽기권한만을 주게 됩니다.

기정으로 닉명의 사용자는 어떠한 등록부에도 쓰기할수 없습니다.

경고 : 만일 FTP봉사기에로의 닉명의 접속을 허용한다면 중요한 자료가 어디에 보관되여있는지를 알게 됩니다.

닉명사용자에 의한 올리적재

닉명의 사용자들이 화일을 올리적재하는것을 허용하기 위하여서는 /var/ftp/pub/안에 쓰기권한만을 허용한 등록부를 만들어야 합니다.

이를 위하여 다음과 같은 지령을 실행합니다.

mkdir /var/ftp/pub/upload

다음 권한(permission)을 변경시키면 닉명의 사용자들은 등록부안의 내용을 현시할수 없습니다.

Chmod 730 /var/ftp/pub/upload

등록부의 긴 목록양식은 다음과 같습니다.

drwx-wx--- 2 root ftp 4096 Feb 13 20:05 upload

경고 : 닉명의 사용자들에게 등록부의 읽기와 쓰기권한을 허용한 관리자들은 자주 자기들의 봉사기가 도적질한 쏘프트웨어의 저장고로 되는것을 발견하게 됩니다.

추가적으로 vsftpd /etc/vsftpd/vsftpd.conf화일에 다음과 같은 행을 추가하여야 합니다.

anon_upload_enables = YES

3) 사용자식별이름

FTP전송은 인증을 요구할 때 보안기능이 없는 망으로 암호화되지 않은 사용자이름들과 통과암호들이 전송되기때문에 체계사용자들이 사용자식별이름으로 봉사기에 접근하지 못하도록 거부하는것이 합리적입니다.

vsftpd에 있는 모든 사용자식별이름을 비활성화시키기 위하여서는 /etc/vsftpd/vsftpd.conf화일에 다음과 같은 행을 추가하면 됩니다.

local_enable = NO

사용자식별이름 제한하기

root사용자와 sudo권한을 가진 사용자들과 같이 특정한 식별이름이나 특정한 식별자그룹을 위한 FTP접근을 비활성화시키기 위하여서는 PAM표 화일을 사용하는것이 제일 쉬운 방법입니다. vsftpd를 위한 PAM설정화일은 /etc/pam.d/vsftpd 입니다.
그것은 매개의 봉사에 대한 사용자식별이름을 직접 비활성화시키는것도 가능합니다.
vsftpd에 있는 특정한 사용자식별이름을 비활성화시키기 위하여서는 /etc/vsftpd.ftpusers에 사용자이름을 추가하면 됩니다.

4) 접근조종을 위하여 TCP Wrapper를 사용하여야 합니다.

FTP 대몬에로의 접근조종을 위하여 TCP Wrapper를 사용하여야 합니다.

5. 우편보안(Securing Sendmail)

sendmail은 다른 MTA들사이에 전자통보문들을 전송하거나 email을 보내기 위한 간단한 우편전송규약을 사용하는 우편전송대행체(MTA : Mail Transfer Agent) 입니다.
많은 MAT들은 암호화된 전송을 진행하지만 대부분은 그렇지 못합니다. 그러므로 어떤 공개된 망으로 email을 보내는것은 전송의 보안기능이 없는 형태로 고찰할수 있습니다.

-봉사거부공격의 제한
전자우편의 본질적특성으로 하여 공격자는 mail을 봉사기에 매우 쉽게 퍼붓고 봉사거부공격을 진행할수 있습니다. /etc/mail/sendmail.mc에 다음과 같은 지령들을 추가함으로써 이와 같은 공격을 제한할수 있습니다.

● confCONNECTION_RATE_THROTTLE – 봉사기가 초당 접수할수 있는 련결의 개수입니다. 기정으로 발신우편은 련결개수의 제한이 없습니다. 만일 이것이 제한되고 이 한계점에 도달하게 되면 련결은 지연됩니다.
● confMAX_DAEMON_CHILDREN – 봉사기에 의하여 생겨날수 있는 child공정의 최대개수입니다. 기정으로 이 개수는 제한되여있지 않습니다. 이러한 제한을 설정하고 이 한계값에 도달하게 되면 련결은 지연

됩니다.

- confMIN_FREE_BLOCKS – 봉사기가 우편을 접수하는데 리용가능한 자유로운 블로크의 최소개수입니다. 기정값은 100입니다.
- confMAX_HEADERS_LENGTH – 통보문머리부의 최대접수크기이다(단위는 Byte).
- confMAX_MESSAGE_SIZE – 한개의 단일통보문의 최대접수크기이다(단위는 Byte).

1) NFS 와 발신우편(Sendmail)

NFS공유구획에 우편입출구완충화등록부 /var/spool/mail/을 만들지 말아야 합니다.

왜냐면 NFSv2과 NFSv3은 사용자와 그룹ID들을 조종하지 않기때문에 2이상의 사용자들이 같은 UID를 가지고 매개의 다른 우편을 접수하고 볼수 있습니다.

주의 : Kerberos를 리용한 NFSv4는 SECRPC_GSS핵심부모형이 UID에 기초한 인증을 리용하지 않기때문에 우와 같은 경우가 없습니다. 그러나 NFS공유구획에 우편입출구완충화 등록부를 배치하지 않는것이 좋을것입니다.

2) 우편사용자만 리용(Mail-only Users)

Sendmail봉사기에서 국부사용자가 공격받는것을 막기 위하여 email프로그람을 사용하여서만 mail사용자들이 Sendmail봉사기에 접근하도록 하는것이 제일 좋습니다. 우편봉사기에서 쉘account들은 허용되지 말아야 하며 /etc/passwd화일에 있는 사용자쉘들은 /sbin/nologin으로 설정되여야 합니다. (root사용자는 례외)

6. 포구조사(Verifying which ports are listening)

망봉사를 설치한 후 체계망대면부에서 실지 동작하고있는 포구들에 주의를 돌리는것이 매우 중요합니다. 임의의 열린 포구는 취약점으로 됩니다.

망에서 동작하고있는 포구들을 목록화하기 위하여서는 2가지 기초적인 수법들이 있습니다. 믿음성이 없는 방법이지만 netstat –an 혹은 Isof –I 와 같은 지령을 사용하는 방법입니다. 이 방법은 이 프로그람들이 망으로부터 콤퓨터에 련결하지 않기때문에 믿음성이 떨어지지만 무엇이 체계에서 동작하고있는가를 검사하기도 합니다. 이러한 리유로 이 프로그람들은 공격자들에 의하여 자주 사용됩니다. 이와 같은 프로그람들의 판본을 갱신하여 리용한다면 망포구들의 상태를 검사하고 또 길을 알수 있을것입니다. 이와 같은 포구주사프로그람으로서는 nmap와 같은것이 있습니다. 망에서 TCP련결된 포구들을 검사하는 지령의 실례를 주었습니다.

nmap –sT –O localhost

이 실례에서 보는바와 같이 sunrpc봉사의 형태는 알수 있지만 834번 포구처럼 알지 못하는 봉사도 있습니다. 봉사목록을 가지고 그에 해당한 봉사형태를 검사할수도 있습니다.

cat /etc/services |grep 834

다음으로 netstat와 Isof지령을 사용하여 포구에 대한 정보를 검사합니다. Netstat를 리용하여 834포구를 검사한 다음과 같은 실례를 보여줍니다.

netstat –anp | gerp 834

tcp 0 0 0.0.0.0:834 0.0.0.0:* LISTEN 653/ypbind

Netstat에서 열린포구의 존재는 크랙커가 이 명령을 리용하여, 침입된 체계에서 포구를 열지는 못하기 때문에 안심할수 있습니다. [p]설정값은 열린포구봉사의 PID(process ID)를 나타냅니다. 이 경우에, ypbind(NIS)에 속하는 열린포구들이 나타나는데 이것은 portmap봉사에 련결되는 RPC봉사를 조종합니다.

isof지령은 봉사들에 대한 열린포구들을 결합할수 있기때문에 netstat와 류사한 정보를 나타냅니다. 그에 대한 실례를 주었습니다.

isof –i | grep 834

ypbind 653 0 7u IPv4 1319 TCP *:834 (LISTEN)

ypbind 655 0 7u IPv4 1319 TCP *:834 (LISTEN)

ypbind 656 0 7u IPv4 1319 TCP *:834 (LISTEN)

ypbind 657 0 7u IPv4 1319 TCP *:834 (LISTEN)

이와 같은 지령들에 대하여 구체적으로 알려면 isof, netstat, nmap에 대한

도움말을 보면 됩니다.

제 3 절 Single Sign-On(SSO)

《붉은별》봉사기용체계 3.0판 SSO기능은 체계탁상형리용자들이 자기들의 암호를 입력하여 가입하는 회수를 감소시킵니다. 몇가지 전문프로그람들은 리용자들이 접속시작화면으로부터 《붉은별》봉사기용체계3.0판에 가입한 다음 그들이 다음번 가입을 할 때 암호를 다시 입력하지 않고 그냥 가입하도록 하는 기능을 수행합니다.

추가적으로 사용자들은 망이 없는 곳에서(offline방식) 혹은 무선접속과 같이 망련결성이 믿음성이 없을 때조차 자기들의 콤퓨터에로 가입할수 있습니다. 망련결성의 믿음성이 없을 때에는 봉사들이 많이 퇴화될것입니다.

1. 지원하는 기능(Supported Applications)

《붉은별》봉사기용체계 3.0판에서 단일화된 가입구성에 의하여 제공된 프로그람들을 보여줍니다.

- Login
- Screensaver
- Firefox

2. 인증방식

《붉은별》봉사기용체계3.0판은 현재 다음과 같은 인증방식을 제공하고 있습니다.

- Kerberos name/password login

3. SSO 의 우점

많은 보안봉사들이 현재 많은 통신규약들과 신용있는 기억장들을 리용합니다. 실례로 SSL, SSH, IPsec, Kerberos등을 포함하고있습니다. 《붉은별》봉사기용체계3.0판의 SSO는 우에서 제시된 요구들을 제공하기 위하여 이것들을 통합하고있습니다.

이것은 Kerberos와 X.509v3인증들을 교체한다는 의미가 아니라 오히려 그것들을 관리하는 관리자들과 체계사용자들의 편리성을 위하여 그것들을 통합하는 수단인것입니다.

제 4 절 PAM(Pluggable Authentication Modules)

체계에 접속하는 사용자들을 허락하는 인증프로그람들은 매 사용자들의 개별적인 식별자를 확인하는 방식으로 인증합니다. 즉 이것은 사용자들이 자기가 누구라는것을 밝히는 과정입니다.

대체로 매 프로그람들은 사용자들을 인증하기 위한 하나의 방법만을 가지고있습니다. 《붉은별》봉사기용체계 3.0 판에서는 많은 프로그람들이 PAM(pluggable authentication modules)라고 부르는 집중형인증체계를 리용하여 설정되여있습니다.

PAM 은 접속할수 있는 모듈구성방식을 사용하는데 이것은 체계관리자가 체계에 인증방책을 설정하는데서 많은 편리성을 가지게 합니다.

대부분 PAM-aware 응용프로그람들을 위한 기정 PAM 설정화일들은 충분하게 설정되여있습니다. 그러나 때때로 PAM설정화일들을 편집할 필요가 있습니다. 그것은 PAM 의 잘못된 설정이 체계보안을 위태롭게 할수 있기때문입니다. 그러므로 무엇보다먼저 이 화일들의 구조를 리해하는것이 중요합니다.

PAM 은 다음과 같은 우점을 가지고있습니다.

- 다양한 응용프로그람들과 함께 사용될수 있는 일반적인 인증체계
- 좋은 유연성과 함께 체계관리자들과 응용프로그람개발자들을 위한 인증을 둘다 조종합니다.
- 하나의 단일화된 완벽한 문서서고를 가지고있는데 이것은 자기의 인증방식을 생성하지 않아도 프로그람을 개발하는 개발자들이 리용가능하게 되여있습니다.

1. PAM 설정화일

/etc/pam.d 등록부에 매개의 PAM-aware 응용프로그람들을 위한 PAM 설정화일들이 있습니다. PAM 의 이전 판본들에는 /etc/pam.conf 화일이

사용되였지만 이 화일은 현재 쓰이지 않으며 다만 /etc/pam.d 등록부가 존재하지 않을 때에만 사용됩니다.

PAM 봉사화일들

매개의 PAM-aware 응용프로그람들과 봉사들은 /etc/pam.d 등록부안에 하나의 화일을 가지고있습니다. 이 등록부에 있는 매 화일은 접근을 조종하기 위한 봉사와 같은 이름으로 되여있습니다.

PAM-aware 프로그람은 자기의 봉사이름을 정의하고 /etc/pam.d 등록부안에 자기의 PAM 설정화일을 설치합니다. 실례로 login 프로그람 은 자기의 봉사이름을 login 으로 정의하며 /etc/pam.d/login 으로 PAM 설정화일을 설치합니다.

2. PAM 설정화일양식

매개의 PAM 설정화일은 다음과 같은 양식으로 되여있습니다.

<module interface> <control falg> <modul name> <module arguments>

1) 모듈대면부(Module Interface)

PAM 모듈대면부의 4가지 형태가 현재 쓰이고있습니다. 이것은 인증공정의 여러가지 측면들을 나타냅니다.

- auth – 이 모듈대면부는 인증을 진행합니다. 실례로 통과암호의 정당성을 요구하고 확증합니다. 이 대면부의 모듈은 그룹성원들과 혹은 Kerberos 표들과 같은 증서들도 설정할수 있습니다.
- account – 이 모듈대면부는 접속이 허락되는가를 검증합니다. 실례로 사용자식별자가 유효한가, 혹은 사용자가 어느시간에 가입하였는가 등을 검사할수 있습니다.
- password – 이 모듈대면부는 사용자통과암호를 바꾸는데 리용됩니다.
- session – 이 모듈대면부는 사용자쎄션들을 설정하고 관리합니다. 또한 사용자의 home 등록부를 mount 하거나 사용자 우편함을 유용하게 만드는것과 같은 접속들을 허용하는것이 필요되는 추가적인 과제들을 수행할수도 있습니다.

주의 : 개별적인 모듈은 임의의 혹은 모든 모듈대면부들을 제공할수 있습니다. 실례로 pam_unix.so 는 모든 4 개의 모듈대면부들을 제공합니다.

PAM 설정화일에서 모듈대면부는 첫번째 마당에서 정의됩니다. 실례로 다음과 같이 설정할수 있습니다.

auth required pam_unix.so

이것은 pam_unix.so 모듈의 auth 대면부를 리용하는 PAM 을 보여줍니다.

묶어진 모듈대면부들

모듈대면부지령들은 하나로 묶어질수 있습니다. 이렇게 함으로써 여러가지 모듈들을 하나의 목적으로 함께 사용할수 있습니다. 모듈조종기발이 "sufficient" 혹은 "requisite" 값으로 사용된다면 모듈들이 목록화되여있는 순서가 인증공정에서 중요합니다.

Stacking(묶기)을 리용하면 관리자가 사용자들의 인증을 허용하기 전에 존재하는 특정한 조건들을 요구하기 쉽게 됩니다. 실례로 reboot 지령은 서 보여주는 바와 같이 일반적으로 여러개의 모듈들을 함께 사용합니다.

```
[root@localhost  ~]# cat /etc/pam.d/reboot
#%PAM-1.0
auth  sufficient  pam_rootok.so
auth  required  pam_조작탁.so
#auth  include  system_auth
account  required  pam_permit.so
```

- 첫번째행은 설명문이므로 공정절차를 나타내지 않습니다.
- auth sufficient pam_rootok.so – 이 행은 pam_rootok.so 모듈이 UID 가 0 인가를 확인함으로써 현재 사용자가 root 인가를 검사하는것을 보여줍니다.
- auth required pam_조작탁.so – 이 행은 사용자가 인증하는것을 시도

하는 pam_조작탁.so 모듈을 리용합니다. 만일 이 사용자가 조작탁에 이미 가입되여있다면 pam_조작탁.so 는 /etc/security/조작탁.apps/ 등록부에 봉사이름(reboot)와 같은 이름으로 화일이 있는가를 검사합니다. 만일 화일이 존재하면 인증은 성공하며 다음 모듈에로 넘어갑니다.

- #auth include system-auth – 이 행은 설명행입니다.
- account required pam_permit.so – 이 행은 root 사용자 혹은 다른 누군가가 체계를 재기동하기 위하여 조작탁에 가입하는것을 허락하는데 리용됩니다.

2) 조종기발

모든 PAM 모듈들은 호출될때 성공 혹은 실패의 결과를 얻어낸다. 조종기발은 PAM 이 어떤 결과를 나타내는가를 말해줍니다. 모듈은 부분적인 순서에 따라 묶어질수 있으며 조종기발은 부분모듈의 성공 혹은 실패가 봉사에로 사용자인증의 종합적인 성공이 얼마나 중요한가를 결정합니다.

조종기발에는 4 가지가 있습니다.

- required – 모듈결과는 다음 공정을 계속하기위하여 인증이 성공되여야 합니다. 만일 여기서 검사가 실패하였다면 그 대면부를 참조하는 모든 모듈검사의 결과가 완성될때까지 사용자에게 통지되지 않습니다.
- requisite – 모듈결과는 다음공정을 계속하기 위하여 인증이 성공되여야 합니다. 그러나 만을 여기서 검사가 실패한다면 사용에게는 처음으로 실패된 required 혹은 requisite 모듈검사를 되돌리는 통보문를 즉시에 통지되지 않습니다.
- sufficient – 모듈결과는 만일 그것이 실패한다면 무시됩니다. 그러나 만일 sufficient 기발을 가진 모듈결과가 성공하고 required 기발을 가진 실패된 이전공정모듈이 없다면 다른 요구결과들이 없으며 사용자는 봉사에 인증됩니다.
- optional – 모듈결과는 무시됩니다. optinal 로 기발설정된 모듈은 다만 그 대면부를 참조하는 다른모듈이 없을 때 성과적인 인증을 위하여 필요하게 됩니다.

이 밖에도 PAM 을 위한 쓸모있는 조종기발들이 나오고있습니다.

Pam.d 에 대한 도움말과 PAM 문서는 /usr/share/doc/pam-<version-number>/ 등록부에 있는데 여기서 <version-number>는 체계에 있는 PAM 의 판본번호입니다.

3) 모듈이름

모듈이름은 특정한 모듈대면부에 있는 삽입가능한 모듈의 이름으로 PAM 을 규정합니다. 《붉은별》봉사기용체계 3.0 판의 이전판본들에서는 모듈의 경로가 PAM 설정화일에 제시되여있없습니다. 그러나 다중서고체계들이 출현함으로서 /lib64/security/ 등록부에 64-bit PAM 모듈들이 보관되며 등록부이름은 생략되였습니다. 왜냐면 응용프로그람은 libpam 의 적당한 판본에 련결되여있는데 이때 모듈의 정확한 판본을 지적할수 있기 때문입니다.

4) 모듈인수

PAM 은 일부 모듈을 위한 인증을 진행하는 기간 삽입가능한 모듈로 정보를 보내기 위한 인수를 사용합니다.

실례로 pam_userdb.so 모듈은 사용자를 인증하기 위하여 KIM DB화일에 보관된 정보를 리용합니다. KIM DB 는 많은 응용프로그람들에 삽입된 공개원천자료기지체계입니다. 그 모듈은 db 인수를 가지고있음으로 하여 Berkeley DB 는 요구되는 봉사를 위하여 사용되는 자료기지가 어느것인가를 알수 있습니다.

실례를 보여주었습니다. 여기서 <path-to-file>은 KIM DB 자료기지화일의 경로입니다.

auth required pam_userdb.so db=<path-to-file>

무효인수들은 일반적으로 무시되며 PAM 모듈의 성공과 실패에 영향을 미치지 않습니다. 그러나 일부 모듈들은 무효인수들에 의하여 실패할수도 있습니다. 대부분의 모듈들은 /var/log/secure 화일에 오유를 리력합니다.

3. PAM 설정화일들의 견본

PAM 응용프로그람 설정화일의 견본을 보여줍니다.

```
#%PAM-1.0
auth   required   pam_securetty.so
auth   required   pam_unix.so   nullok
auth   required   pam_nologin.so
account   required   pam_unix.so
password  required   pam_cracklib.so   retry=3
password  required   pam_unix.so   shadow  nullok   use_authtok
session   required   pam_unix.so
```

- 첫행은 #표식으로 시작된 설명문입니다.
- 두번째부터 네번째까지의 세 모듈들은 인증가입을 위한 묶음입니다.
  - auth required pam_securetty.so – 이 모듈은 만일 사용자가 root 로 가입한다면 사용자가 가입한 tty 가 /etc/securetty 화일(이 화일이 존재하면)에 목록화되는것을 보호합니다.

    만일 이 화일에 tty가 목록화되여있지 않으면 Login incorrect 통보문으로 root 로써의 가입의 실패를 나타냅니다.
  - Auth required pam_unix.so nullok – 이 모듈은 사용자들의 통과암호를 요구하며 /etc/passwd 와 /etc/shadow(만일 존재한다면)에 보관된 정보를 리용하여 통과암호를 검사합니다.

    인수 nullok 는 pam_unix.so 모듈이 빈통과암호를 허용하도록 하는 기능을 수행합니다.
- auth required pam_nologin.so – 이것은 인증의 마지막 단계입니다. /etc/nologin 화일이 존재하는가를 검사합니다. 만일 존재하고있으며 사용자가 root 가 아니면 인증은 실패합니다.

  주의 : 이 실례에서는 첫번째 auth 모듈이 실패하였어도 모든 3 개의 auth 모듈들이 검사되였습니다. 이것은 사용자가 자기의 인증이 어느단계에서 실패하였는가를 알수 없게 합니다. 이것은 공격자들에게 기회를 제공하게 되며 또한 체계를 공격하기 위한 방법을 더 쉽게 찾을수 있게 됩니다.

- account required pam_unix.so – 이 모듈은 임의의 필요한 사용식별자확인을 진행합니다. 실례로 shadow 통과암호들이 능동화되였다면 pam_unix.so 모듈의 account 대면부는 사용자식별자가 탈퇴되였는가, 혹은 제정된 기간안에 통과암호를 바꾸지 않았는가 하는것을 검사합니다.
- password required pam_cracklib.so retry=3 – 만일 통과암호가 기한이 지났다면 pam_cracklib.so 모듈의 통과암호구성부분은 새 암호를 요구합니다. 다음 새로 설정된 암호가 사전에 기초한 통과암호해득프로그람으로 쉽게 풀리지 않는가를 검사합니다.
  - 인수 retry=3 은 만일 검사가 첫번째에서 실패하였다면 사용자가 강한 암호를 설정할수 있는 기회를 2 번 더 주게 됩니다.
- password required pam_nuix.so shadow nullok use_authtok – 이 행은 다음과 같은 내용을 담고있습니다. 즉 만일 프로그람이 사용자의 통과암호를 바꾼다면 그를 위하여 pam_unix.so 모듈의 password 대면부를 사용하여야 합니다.
  - 인수 shadow 는 사용자의 통과암호가 갱신될 때 shadow 통과암호를 생성하기 위한 모듈을 나타냅니다.
  - 인수 nullok 는 사용자가 빈 통과암호로 자기의 통과암호를 바꾸는것을 허용하도록 모듈을 설정합니다.
  - 이행의 마지막 인수인 use_authtok 는 PAM 모듈을 묶을 때 순차의 중요성을 보여주는 좋은 실례로 됩니다. 이 인수는 모듈이 사용자들에게 새로운 통과암호를 재촉하지 않게 합니다. 대신에 이것은 이전 통과암호모듈에 의하여 기록된 임의의 통과암호를 수락합니다.이 방법에 의하여 모든 새로운 통과암호들이 접수되기전에 보안 통과암호를 위한 pam_cracklib.so 검사를 통과하여야 합니다.
- session required pam_unis.so – 마지막 행은 session 을 관리하기 위한 pam_unix.so 모듈의 session 대면부를 나타냅니다. 이 모듈은 사용자이름과 봉사형태를 /var/log/secure 에 매 쎄션의 시작과 끝에서 리력합니다. 이 모듈은 그것을 추가적인 기능을 위한 다른 쎄션모듈들과 결합함으로써 보충될수 있습니다.

4. PAM 모듈의 생성

PAM-aware 응용프로그람을 리용하는 임의의 시간에 PAM 모듈을 생성하거나 추가할수 있습니다.

실례로, 개발자는 단일회수통과암호생성방법을 만들고 PAM 모듈에 추가할수 있습니다. PAM-aware 프로그람들은 즉시 새로운 모듈과 통과암호를 재콤파일하거나 다른 변경이 없이 사용할수 있습니다.

이것은 개발자들과 체계관리자들이 재콤파일이 없이 여러가지 프로그람들을 위한 인증방법을 실험하는데 리용됩니다. 문서는 /usr/share/doc/pam-<version-number>/등록부안에 작성되여있습니다. 여기서 <version-number>는 체계에 있는 PAM 의 판본번호입니다.

5. PAM 과 관리자증서 숨기기

《붉은별》봉사기용체계 3.0 에서 많은 관리도구들은 pam_timestamp.so 모듈을 리용하여 5 분안에 사용자들의 권한을 높여 줍니다. 그것은 어떻게 이 도구가 동작하는가를 리해하는것이 중요합니다. 왜냐면 pam_timestamp.so 가 있는 말단으로부터 동작하는 사용자들이 조작탁에 물리적으로 접근하는 누군가에 의하여 조작되여 이것을 여는 효과를 가지기 때문입니다.

PAM 시간계수체계에서는 도형방식관리응용프로그람이 사용자가 root 통과암호로 가입하도록 요구합니다. 사용자가 인증되면 pam_timestamp.so 모듈은 시간계수화일을 생성합니다. 기정으로 이것은 /var/run/sudo/ 등록부에 생성됩니다. 만일 시간계수화일이 이미 존재하면 도형방식 관리프로그람들은 통과암호를 요구하지 않습니다. 대신에 pam_timestamp.so 모듈은 시간계수화일을 갱신하고 사용자에게 추궁을 하지않는 관리적접근의 보충적인 5 분이라는 시간을 줍니다.

/var/run/sudo/<user> 화일을 조사함으로써 시간계수화일의 실제상태를 검증할수 있습니다. 화면상에는 그 화일이 unknown:root 로 나타납니다. 만일 그것이 현시되고 그의 시간계수가 5 분보다 작으면 증서들이 타탕한것으로 됩니다.

시간계수화일의 존재는 통로의 공시구역에 인증아이콘으로 나타납니다.

1) 시간계수화일 삭제하기

PAM 시간계수기가 실행되는 조작탁을 끄기전에 그것은 시간계수화일이 파괴되는가를 검사합니다.

PAM 시간계수화일에 대하여 다음과 같은 내용을 참작하여야 합니다.

- 만일 ssh 를 리용하여 원격으로 체계에 가입하였다면 /sbin/pam_timestamp_check –k root 지령을 사용하여 시간계수화일을 파괴하여야 합니다.
- 또한 권한을 가진 응용프로그람을 실행하고있는 같은 말단 창문에서 /sbin/pam_timestamp_check –k root 지령을 실행하여야 할 필요가 있습니다.
- /sbin/pam_timestamp_check –k 지령을 리용하기 위하여 pam_timestamp.so 모듈을 원래부터 불러들이는 사용자로써 가입하여야 합니다. 이 지령을 리용하기 위하여 root 로 가입하지 말아야 합니다.
- 만일 화면상에 증서들을 끄고 싶다면 다음과 같은 지령을 실행하여야 합니다.

```
/sbin/pam_timestamp_check -k root </dev/null >/dev/null 2>/dev/null
```

이 지령의 실행이 실패하면 모든 증서들이 지령을 실행하고있는 pty 로부터 삭제될것입니다.

2) 일반적인 pam_timestamp 지령들

Pam_timestamp.so 모듈은 여러가지 지령들을 가지고있습니다. 2 개의 가장 대표적인 지령들을 소개합니다.

- timestamp_timeout – 시간계수화일이 효력을 가지는 기간을 초단위로 확정합니다. 기정값은 300 이다(5 분).
- timestampdir – 시간계수화일이 보관되는 등록부를 확정합니다. 기정값은 /var/run/sudo/입니다.

6. PAM 과 장치소유권(Device Ownership)

《붉은별》봉사기용체계 3.0 판에는 콤퓨터의 물리적조작탁에 가입한 첫번째 사용자가 정해진 장치를 조종할수 있으며 root 사용자로서 진행하는 일반적인 정해진 과제를 수행할수 있습니다. 이것은 pam_조작탁.so 라고 부르는 PAM 모듈에 의하여 조종됩니다.

1) 장치소유권

사용자가《붉은별》봉사기용체계 3.0 판 체계에 가입하면 pam_조작탁.so 모듈은 login 에 의하여 호출됩니다. 만일 이 사용자가 조작탁에 처음으로 가입하는 사용자라면 (조작탁 사용자)그 모듈에 의하여 사용자가 root 로서 일반적으로 소유하고있는 여러가지 장치들의 소유권을 가지게 됩니다. 그 조작탁 사용자는 그 사용자가 끝내는 마지막 국부쎄션까지 이 장치들을 소유하게 됩니다. 이 사용자가 가입탈퇴한후에 장치들의 소유권은 root 사용자에게로 다시 되돌아갑니다.

장치들에는 음성카드, floopy 디스크구동기, CD-ROM 구동기등이 포함되는데 이것으로 제한되는것은 아닙니다.

국부사용자는 root 접근이 없이도 이 장치들을 조종할수 있습니다. 그러므로 조작탁사용자는 간단한 일반 업무를 진행할수 있습니다.

다음과 같은 화일들을 편집함으로써 pam_consloe.so 에 의하여 조종되는 장치의 목록을 작성할수 있습니다.

- /etc/security/조작탁.perms
- /etc/security/조작탁.perms.d/50-default.perms

우의 화일들에서 기정으로 설정되여있는 내용을 편집하여 여러가지 장치들을 추가하거나 변경시킬수 있습니다. 이때 50-default.perms 화일을 변경시키기 보다는 오히려 새로운 화일을 생성하고 거기에 요구되는 항목을 입력하는것이 좋을것입니다.(실례로 xx-name.perms) 새롭게 작성하는 화일은 이름이 50 보다 큰 수자로 설정되여야 합니다. (실례로 51-default.perms)이것은 기정으로 있는 50-default.perms 화일을 갱신할것입니다.

2) 응용프로그람 접근

조작탁 사용자는 /etc/security/조작탁.apps/등록부안에 사용하기 위한 적당한 프로그람들을 설정함으로써 그에 대한 접근도 진행할수 있습니다.

이 등록부 안에는 /sbin과 /usr/sbin등록부에 있는 응용프로그람들을 실행할수 있도록 조작탁사용자를 허용하도력 하는 설정화일들이 들어있습니다.

이 설정화일들은 설치되여있는 응용프로그람들과 같은 이름으로 되여있습니다.

다음과 같은 실례에서 보여주는 바와 같이 조작탁사용자가 체계의 끄기나 재기동과 같은 3개의 프로그람들에 접근할수 있습니다.

- /sbin/halt
- /sbin/reboot
- /sbin/poweroff

이것들은 PAM-aware 응용프로그람들이기 때문에 그것들은 사용할때 pam_조작탁.so 모듈을 호출한다

7. 추가자원들

서 서술되는 자원들은 PAM을 설정하고 사용하기 위한 상세한 방법들을 보여줍니다. 이 자원들의 지원으로 체계에서 PAM 설정화일들을 어떻게 구성되여있는가를 더욱 쉽게 리해할수 있습니다.

설치된 PAM 문서화

- PAM-관련도움말페지 – PAM에 대한 여러가지 응용프로그람들과 설정화일들을 위하여 여러개의 도움말페지들이 존재합니다.

  설정화일들

  - pam – PAM설정화일들의 구조와 목적을 포함하는 PAM에 대한 소개적인 정보가 있습니다. 이 페지들은 /etc/pam.conf 와 /etc/pam.d/등록부에 있는 개별적인 설정화일들에 서술되여있습니다. 기정으로 《붉은별》봉사기용체계 3.0판 는 /etc/pam.d/ 등록부에 있는 개별적인 설정화일들을 리용하며 만일 그것이 존

재하여도 /etc/pam.conf 는 무시합니다.

- pam_조작탁 – pam_조작탁.so 모듈의 목적을 나타냅니다. 또한 PAM 설정화일에 입력하는 문장구성법을 나타내기도 합니다.
- 조작탁.apps - /etc/security/조작탁.apps 설정화일안에 쓸모있는 형식들과 기능들을 나타내는데 이것은 PAM 에 의하여 설계된 조작탁 사용자가 접근할수 있는 응용프로그람을 정의합니다.
- 조작탁.perms - /etc/security/조작탁.perms 설정화일안에 쓸모있는 형식들과 기능들을 나타내는데 이것은 PAM 에 의하여 설계된 조작탁 사용자 허락을 특정화합니다.
- pam_timestamp – pam_timestamp.so 모듈을 나타냅니다.

- /usr/share/doc/pam-<version-number> - 여기에 대하여서는 많은 문서들에 나와 있습니다.
- /usr/share/doc/pam-<version-number>/txts/README.pam_timestamp – 여기서는 pam_timestamp.so PAM 모듈에 대한 정보가 있습니다. 여기서 <version-number>는 PAM 의 판번호입니다.

제 5 절 TCP Wrapper 와 xinetd

망봉사들에로의 접근을 조종하는것은 봉사기관리자들에게 있어서 가장 중요한 보안문제입니다.《붉은별》봉사기용체계 3.0 판은 이러한 기능을 수행하는 여러가지 도구들을 제공합니다. 실례로, iptables 에 기초한 방화벽이 있는데 이 우점은 망파케트들을 려과합니다. 그를 위한 망봉사기로서 TCP Wrapper 는 “wrapped”망봉사들에 접속을 허락하지 않거나 호스트들이 정의되는 보안추가계층을 추가합니다. 매 개별적인 망봉사는 xinetd 대형봉사기입니다. 이 봉사는 망봉사들의 부분설정에로의 련결을 조종하고 더우기 접근조종을 진행하므로 대형봉사기라고 합니다.

여기서는 망봉사들에 대한 접근조종에서 TCP Wrapper 와 xinetd 의 규칙들과 이도구들에 대한 가입과 리용, 관리에서 나서는 몇가지 문제들에 대하여 서술합니다.

1. TCP Wrapper

TCP Wrapper 패키지(tcp_wrappers)는 기정으로 설치되여있으며 망봉사

들에 대한 host 준위의 접근조종을 제공합니다. 패키지에서 가장 중요한 부분품은 /usr/lib/libwrap.a 서고입니다.

TCP-Wrappers 봉사에 련결을 시도하면 봉사는 우선 의뢰기가 허용되였는가를 결정하기 위하여 host 의 접근화일(/etc/hosts.allow 와 /etc/hosts.deny)를 확인합니다. 대부분의 경우에 /var/log/secure 혹은 /var/log/messages 에 요구되는 봉사와 요구하는 의뢰기의 이름을 쓰기 위하여 syslog 대몬을 리용합니다.

만일 의뢰기가 접속이 허용되였다면 TCP Wrapper 는 요구한 봉사에로의 접속을 조종하며 의뢰기와 봉사기사이의 통신을 진행합니다.

접근조종과 가입을 위하여 추가적으로 TCP Wrapper 는 요구한 망봉사에로의 련결에 대한 조종을 거부하거나 풀어놓기전에 의뢰기와 호상작용하기 위한 지령들을 실행합니다.

TCP Wrapper 는 봉사기관리자들을 위한 보안도구들의 집합체이므로 《붉은별》봉사기용체계 3.0 판에서의 대부분의 망봉사들은 libwrap.a 서고에 련결되여있습니다. /usr/sbin/sshd, /usr/sbin/sendmail, /usr/sbin/xinetd 를 포함하는 일부 개별적인 응용프로그람들이 있습니다.

주의 : 만일 망봉사가 libwrap.a 에 련결되였는가를 알아보려면 다음과 같은 지령을 실행하면 됩니다.

```
ldd <binary-name> | grep libwrap
```

여기서 <binary-name>은 망봉사 binary 의 이름입니다. 만일 지령실행후 출력이 없으면 망봉사는 그와 련결이 되지 않은 상태입니다. /usr/sbin/sshd 가 libwrap.a 와 련결된 실례를 보여주었습니다.

```
[root@localhost ~]# ldd/usr/sbin/sshd | grep libwrap
Libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@localhost ~]#
```

TCP Wrapper 의 우점

TCP Wrapper 는 망봉사조종기술을 위한 다음과 같은 우점들을 가지고있습니다.

- 의뢰기와 망봉사사이의 투명성 – 의뢰기와 봉사기(wrapped) 사이의 련결은 TCP Wrapper 가 사용되고있다는것을 알지 못합니다. 합

법적인 사용자들은 요구되는 봉사에 가입을 하고 련결을 진행합니다.

- 복합통신규약의 중심적관리 – TCP Wrapper 는 그것이 보호하고있는 망봉사들로부터 개별적으로 조작합니다. 이때 접근조종설정화일들의 일반적인 모임을 공유하기 위하여 많은 봉사기응용프로그람들이 허용되여있으며 이로 하여 관리가 보다 편리합니다.

2. TCP Wrapper 설정화일

의뢰기가 봉사에로의 련결이 허용되는가를 알아보기 위하여 TCP Wrapper 는 호스트접속화일로서 일반적으로 제공하는 다음과 같은 2 개의 화일을 보여줍니다.

- /etc/hosts.allow
- /etc/hosts.deny

TCP-wrapped 봉사는 의뢰기요구를 접수한후 다음과 같은 단계를 실행합니다.

- /etc/hosts.allow – TCP-wrapped 봉사는 련속적으로 /etc/hosts.allow 화일을 분석하고 봉사를 위한 규칙들을 설정합니다. 만일 규칙과 맞으면 그것은 접속을 허용하고 그렇지 않으면 다음 단계로 넘어갑니다.
- /etc/host.deny – TCP-wrapped 봉사는 /etc/hosts.deny 화일을 분석합니다. 이때 규칙에 맞으면 련결을 거부하고 그렇지 않으면 봉사에로 접속을 진행합니다.

망봉사들을 보호하기 위한 TCP Wrapper 사용에서 고려하여야 할 중요한문제점들이 있습니다.

- hosts.allow 화일에 있는 접속규칙들이 먼저 확인되기 때문에 이것은 hosts.deny 보다 앞선 공정으로 됩니다. 그러므로 만일 hosts.allow 에서 봉사의 접속이 허용되면 hosts.deny 에 있는 거부규칙들은 무시됩니다.
- 매 규칙화일을 읽을 때 우에서부터 아래로 검색하며 주어진 봉사에 대한 첫 일치된 규칙은 한번만 리행됩니다. 그러므로 규칙의 순서를 잘 정하는것이 매우 중요합니다.

- 만일 봉사에 대한 규칙이 존재하지 않으면 또는 화일이 존재하지 않으면 봉사에로의 접속은 인정됩니다.
- TCP-Wrapped 봉사들은 호스트접속화일들로부터 규칙들을 캐쉬하지 않습니다. 그러므로 hosts.allow 나 hosts.deny 의 그 어떠한 변화도 봉사를 재기동함이 없이 즉시에 효력을 나타냅니다.

경고 : 만일 호스트접근화일의 마지막행이 빈행(Enter 건을 누름으로서 생성되는 행)이 아니라면 마지막규칙이 실패로 되며 오유가 /var/log/messages 혹은 /var/log/secure/에 보관됩니다. 이와 같은 실례를 보여줍니다.

warning: /etc/hosts.allow, line 20: missing newline or line too long

1) 접근규칙형식화

/etc/hosts.allow 와 /etc/hosts.deny 에 대한 형식은 같습니다. 매 규칙은 자기의 행을 가져야 하며 빈행들이나 "#"로 시작되는 행은 무시됩니다.

매 규칙은 다음과 같은 기본적인 형식을 리용합니다.

<daemon list>:<client list> [: <option>: <option>: …]

- <daemon list> - 처리이름들이나 혹은 모든 wilecard(봉사이름이 아님)들이 반점으로 구분되여 목록화되여있습니다. 대몬목록은 매우 편리한 조작들을 진행할수 있습니다.
- <client list> - 호스트이름들, 호스트IP 주소, 특정한 패턴 확장카드들이 반점으로 구분하여 목록화되여있습니다.
- <option> - 규칙들을 설정할때 설정항목 혹은 두점으로 구분된 항목목록을 나타냅니다. 설정마당들은 확장, 쉘지령의 실행, 접근의 허가와 거부, 가입절차의 변경등을 지원합니다.

호스트접근규칙의 기초적인 실례를 보여줍니다.

vsftpd: .example.com

이 규칙인 example.com 령역에 있는 임의의 호스트로부터 FTP 대몬에로(vsftpd)련결을 보기위한 TCP Wrapper 를 설정합니다. 만일 규칙이

host.allow 로 되여있으면 련결은 접수됩니다. 만일 hosts.deny 로 나타나면 련결은 거절됩니다.

다음과 같은 실례에서 보여준 접근규칙은 보다 더 구체적이며 2 개의 설정마당을 리용합니다.

Sshd : .example.com \ : spawn /bin/echo '/bin/date' access denied>>/var/log /sshd.log \ : deny

여기서 매 설정마당은 "\"에 의하여 구분됩니다.

실례규칙은 example.com 령역에서 한 호스트로부터 SSH 대몬에로 련결을 시도하는 상태를 보여주는데 이때 echo 지령을 실행하면 특정한 리력화일이 생겨나며 련결은 거부됩니다. 그러므로 deny 설정이 사용되면 이 행은 hosts.allow 화일에서 나타났다고 하여도 접근을 거부합니다.

확장카드

확장카드는 TCP Wrapper 가 대몬들이나 호스트들의 그룹들을 더 쉽게 정합시키게 합니다. 그것들은 접근규칙의 의뢰기목록마당에서 대부분 사용됩니다.

- ALL - 모든것이 정합됩니다. 이것은 대몬목록과 의뢰기목록에서 둘다 리용될수 있습니다.
- LOCAL – 국부호스트와 같이 "."이 없는 임의의 호스트들을 정합합니다.
- KNOWN – 호스트이름과 호스트주소를 알고있거나 사용자를 알고있는 임의의 호스트를 정합합니다.
- UNKNOWN – 호스트이름이나 호스트주소를 모르거나 사용자를 모르는 호스트들을 정합합니다.
- PARANOID – 호스트이름이 호스트주소와 정합되지 않는 임의의 호스트를 정합합니다.

패턴

패턴들은 의뢰기호스트들의 그룹을 더 정확히 특정화하기 위한 접근규칙들의 의뢰기마당에서 리용됩니다.

의뢰기마당에 들어가는 일반적인 패턴들의 목록을 보여줍니다.

- "."으로 시작하는 호스트이름 – 호스트이름의 시작을 "."으

로 하면 그것은 그 이름의 목록화된 구성부분들을 공유한 모든 호스트들을 정합합니다. 실례로 서 보여준 바와 같이 이것은 example.com 령역에 있는 임의의 호스트를 접수합니다.

- ".”으로 끝난는 IP주소 - 점으로 끝나는 IP주소는 IP주소의 첫 주소를 공유하는 모든 호스트들을 정합합니다. 다음과 같은 실례에서는 192.169.x.x 망의 임의의 호스트를 접수하는 경우입니다.

ALL: 192.168.

- IP 주소/네트마스크 – 네트마스크는 IP 주소들의 부분적인 모임들에로의 접근을 조종하기 위한 패턴으로도 사용될수 있습니다. 다음과 같은 실례에서 보는바와 같이 이것은 192.168.0.0 부터 192.168.1.255 의 령역에 해당한 주소에 대한 임의의 호스트를 접수합니다.

ALL:192.168.0.0/255.255.254.0

- [IPv6 주소]/prefixlen – [net]/prefixlen 쌍은 IPv6 주소들의 부분적인 그룹에로의 접근을 조종하기 위한 하나의 패턴으로서 사용될수도 있습니다. 다음과 같은 실례에서는 3ffe:505:2:1:: 부터 3ffe:505:2:1:ffff:ffff:ffff:ffff:령역의 주소에 있는 임의의 호스트에로 접수됩니다.

ALL:[3ffe:505:2:1::]/64

- 별표(*) – 별표는 호스트이름이나 IP 주소들의 전체그룹을 정합시키는데 리용됩니다. 다음과 같은 실례는 example.com 령역의 임의의 호스트를 허용하는 실례이다

ALL:*.example.com.

- 사선(/) – 만일 의뢰기목록이 슬래쉬로 시작된다면 그것은 화일이름과 같이 관리됩니다. 이것은 호스트들의 많은 량의 규칙들이 필요할때 유용합니다. 다음과 같은 실례는 모든 Telnet 련결을 위한 /etc/telnet.hosts 화일에 대한 TCP Wrapper 를 나타냅니다.

In.telnetd:/etc/telnet.hosts

Portmap 와 TCP Wrapper

TCP Wrapper 의 Portmap 대면부는 호스트현시를 제공하지 않습니다. 이것은 portmap 는 호스트들을 알아보기 위한 호스트이름을 사용할수 없다는것입니다. 그러므로 hosts.allow 나 hosts.deny 에 있는 portmap 를 위한 접근조종규칙들은 IP 주소나 열쇠단어 ALL 을 사용하여야 합니다.

Portmap 접근조종규칙의 변화는 즉시 효과가 나타나지 않습니다. 즉 portmap 봉사를 재기동하여야 합니다.

NIS 와 NFS 와 같은 광범히 리용되는 봉사들은 조작에서 portmap 에 의존하므로 이 제한점들에 주의를 돌려야 합니다.

조종자(operator)

현재 접근조종규칙들은 하나의 조종자 EXECPT 를 포함합니다. 그것은 대몬목록과 의뢰기규칙목록에서 다 사용될수 있습니다.

EXCEPT 조종자는 같은규칙에서 정합되는것외에 허용합니다.

다음과 같은 hosts.allow 화일에 대한 실례는 모든 example.com 호스트들이 cracker.example.com 을 제외한 모든 봉사들에로의 련결을 허용합니다.

ALL:.example.com EXCEPT cracker.example.com

서 hosts.allow 화일에 대한 다른 실례는 보여주는데 이것은 192.168.0.x 망마스크의 의뢰기들은 FTP 를 제외한 모든 봉사를 리용할수 있습니다.

ALL EXCEPT vsftpd:192.168.0.

2) 설정마당

추가적으로 접근을 허용하거나 거부하기 위한 규칙을 위하여 TCP Wrapper 의 《붉은별》봉사기용체계 3.0 판대면부는 설정마당을 통한 접근조종언어를 지원하고있습니다. 호스트접근규칙에서 설정마당들을 사용함으로써 관리자는 접근조종통합이나 쉘지령의 실행과 같은 여러가지 업무들을 수행할수 있습니다.

가입

설정마당은 관리자들이 severity 를 사용함으로써 가입을 쉽게 할수 있도록 합니다.

SSH 대몬으로 접속하는 실례는 다음과 같습니다.

sshd : .example.com:severityemerg

접근조종

설정마당은 관리자들이 호스트들이 접속을 허용하거나 거부하는것을 조종할수도 있게 합니다.

다음의 실례와 같이 client-1.example.com 은 SSH 련결이 허용되며 client -2.example.com 은 련결이 거부됩니다.

sshd : client-1.example.com : allow

sshd : client-2.example.com : deny

이 과정을 화일로 관리할수 있는데 그 화일들은 hosts.allow 혹은 hosts.deny 입니다. 일부 관리자들은 이 방법이 접근규칙들을 정의하는데서 더 쉬운 방법으로 여기고있습니다.

쉘지령

설정마당은 접근규칙들이 쉘지령을 실행하도록 허용합니다.

- Spawn – 자식공정과 같이 쉘지령을 실행합니다. 이 지령은 /usr/sbin/safe_finger 를 리용하는것과 같이 과제를 수행할수 있습니다. 이것은 의뢰기가 요구하는것에 대한 더 많은 정보를 얻거나 echo 지령을 리용하여 특별한 리력화일을 생성하기 위하여 리용됩니다.

  in.telnetd : .example.com \

  :spawn /bin/echo '/bin/date' from %h>>/var/log/telnet.log \

  :allow

- Twist – 요구되는 봉사를 교체합니다. 이 지령은 공격자들을 잡기 위하여 자주 사용되는 지령입니다. (일명 《꿀단지》라고도 합니다.) 그것은 련결된 의뢰기들에 통보문들을 보내는데도 사용될수 있습니다.
- vsftpd : .example.com \
- : twist /bin/echo "Access denied!"

보충적인 지령들

이것들은 spawn 과 twist 지령과 같이 사용되면서 의뢰기나 봉사기 프로쎄스들에 대한 정보를 얻어내는 역할을 합니다.

이에 대한 항목들을 서술합니다.

- %a – 의뢰기의 IP 주소를 되돌립니다.
- %A – 봉사기의 IP 주소를 되돌립니다.
- %c – 사용자이름, 호스트이름 사용자이름과 IP 주소와 같은 의뢰기의 정보를 되돌립니다.
- %d – 대몬프로쎄스이름을 되돌립니다.
- %h – 의뢰기의 호스트이름을 되돌립니다. (호스트이름을 모르는 경우 IP 주소를 되돌립니다.)
- %H – 봉사기의 호스트이름을 되돌립니다. (호스트이름을 모르는 경우 IP 주소를 되돌립니다.)
- %n – 의뢰기의 호스트이름을 되돌립니다. 만일 모르는 경우 unknown 을 현시하고 의뢰기의 호스트이름과 호스트 주소가 정합되지 않으면 paranoid 를 현시합니다.
- %N – 봉사기의 호스트이름을 되돌립니다. 만일 모르는 경우 unknown 을 현시하고 봉사기의 호스트이름과 호스트주소가 정합되지 않으면 paranoid 를 현시합니다.
- %p – 대몬의 프로쎄스 ID 를 되돌립니다.
- %s – 대몬프로쎄스, 봉사기의 호스트 혹은 IP 주소와 같은 봉사기의 여러가지 형태의 정보를 되돌립니다.
- %u – 의뢰기의 사용자이름을 되돌립니다. 만일 모르는경우 unknown 을 현시합니다.

다음과 같은 실례들은 spawn 지령과 련결하여 추가적인 지령들을 사용하여 얻을수 있는 표준규칙들을 보여줍니다. 이를 통하여 적재된 리력화일에서 의뢰기호스트를 식별할수 있습니다.

example.com령역에 있는 한 호스트로부터 SSH대몬에 의한 련결이 시도되였다면 이 시도를 보관하기 위하여 echo 지령이 실행되며 이때 %h 를 리용하여 의뢰기의 호스트이름이 보관됩니다.

이에 대한 구체적인 정보를 얻으려면 hosts_access(man 5 hosts_access)의 도움말과 hosts_options 의 도움말을 참고하면 됩니다.

3. xinetd

xinetd 대몬은 FTP, IMAP, Telnet 와 같은 일반적인 망봉사들의 부분모임

에로의 접근을 조종하는 TCP-wrapped super 봉사입니다. 그것은 접근조종, 강제가입, 속박, 재련결, 원천리용조종을 위한 봉사특정설정항목들도 제공합니다.

의뢰기가 xinetd 에 의하여 조종되는 망봉사에 련결을 시도할때 super 봉사는 요구를 접수하고 TCP Wrapper 접근조종규칙들을 검사합니다.

만일 접근이 허용되면, xinetd 는 그 봉사에 대한 자기의 접근규칙하에서 련결을 허용합니다. 련결이 실행된후 xinetd 는 의뢰기와 봉사기사이의 통신에서 더 이상 그어떠한 부분도 담당하지 않습니다.

4. xinetd 구성화일

xinetd 를 위한 구성화일들은 다음과 같습니다.

- /etc/xinetd.conf – 포괄적인 xinetd 설정화일입니다.
- /etc/xinetd.d/ - 이 등록부에는 모든 봉사설정화일들이 있습니다.

1) /etc/xinetd.conf 화일

/etc/xinetd.conf 화일에는 xinetd 의 조종하에 있는 모든 봉사들에 작용하는 일반적인 구성설정항목들이 있습니다. Xinetd 봉사를 처음 시작할 때 이 화일을 읽습니다. 그러므로 이 화일을 변화시킨 다음에는 봉사를 재기동하여야 합니다. /etc/xinetd.conf 화일의 실례를 보여줍니다.

```
Defaults
{
    instaces=60
    log_type=SYSLOGauthpriv
    log_on_failure=HOST
    log_on_success=HOSTPID
    cps =2530
}
```

- instances – xinetd 가 동시에 처리할수 있는 요구의 최대 개수를 특정합니다.
- log_type - /var/log/secure 화일에 리력을 쓰기하는 authpriv 리력

을 사용하기 위한 xinetd 를 구성합니다. FILE /var/log/xinetdlog 와 같이 직접 추가하여 /var/log/등로ㅈ부에 xinetdlog 라고 부르는 리력화일을 생성합니다.

- log_on_success – 성공적인 련결을 리력하는 xinetd 를 구성합니다. 기정으로 원격호스트의 IP 주소와 프로쎄스 ID 가 기록되여 있습니다.
- cps – 임의의 주어진 봉사에 매 초당 최대로 25 개의 련결을 허용하도록 xinetd 를 구성합니다. 만일 이 한계를 넘으면 30 초동안 대기합니다.

2) /etc/xinetd.d/ 등록부

이 등록부에는 봉사에 관계되는 이름과 xinetd 에 의하여 관리되는 매 봉사들을 위한 설정화일들이 들어 있습니다. xinetd.conf 에서 이 등록부는 xinetd 봉사가 시작될 때에만 읽어집니다. 어떠한 변화를 진행한 다음에는 xinetd 봉사를 재기동하여야 합니다.

이 등록부의 화일형식은 /etc/xinetd.conf 와 같이 같은 형식을 사용합니다.

이 화일들이 어떻게 구성되여있는가를 알기 위하여 /etc/xinetd.d/krb5-telnet 화일을 고찰해 보자.

```
service telnet
{
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/Kerberos/sbin/telnetd
log_on_failure += USERID
disable = yes
}
```

제 6 절 Kerberos

망봉사에 대한 사용자인증은 규약이 리용하는 방법이 안전하지 못할 때 전통적인 FTP 와 Telnet 규약들을 리용하여 망을 통하여 암호화되지 않은 통과단어들의 전송이 증명해주는것처럼 위험성을 확인할수 있습니다.

Kerberos 는 안전치 못한 인증방법들을 허용하는 규약들에 대한 요구를 제거하는 방법이며 그에 의하여 전반적인 망보안을 강화합니다.

1. Kerberos 란 무엇인가?

Kerberos는 MIT가 만든 망인증규약이며 망봉사들에 대한 사용자인증을 위해 대칭열쇠암호를 리용합니다. 이것은 통과단어들이 망을 통하여 사실상 절대로 보내지지 않는다는것을 의미합니다.

결과적으로 사용자들이 Kerberos 를 리용하여 망봉사들을 인증할때 망통화량을 감시하여 통과단어들을 수집하려고 시도하는 권한없는 사용자들을 효과적으로 차단할수 있습니다.

1) Kerberos 의 우점

대부분의 관습적인 망봉사들은 통과단어에 기초한 인증방식을 리용합니다. 그러한 방식들은 망봉사기에 자기의 사용자이름과 통과단어를 주어 사용자인증을 요구합니다. 불행하게도 많은 봉사들에서 인증정보의 전달은 암호화되지 않습니다. 보안되여야 할 그러한 방식에 대하여 망은 외부자들에게 접근불가능해야 하며 망상의 모든 콤퓨터들과 사용자들은 신뢰되여야 하며 신뢰받을수 있어야 합니다.

그러한 경우에조차도 인터네트와 련결된 망은 더이상 안전하다고 가정할수 없습니다. 망에 대한 접근을 얻은 공격자는 파케트스니퍼라고도 하는 간단한 파케트분석기를 리용하여사용자이름과 통과단어들을 가로챌수 있으며 사용자계정들과 전체보안인프라구조의 완정성을 위태롭게 합니다.

Kerberos 의 주요설계목적은 망을 통하여 암호화되지 않은 통과단어의 전송을 없애는것입니다. 적절하게 리용된다면 커버로스는 아파케트

스니퍼가 망에서 취하는 위협을 효과적으로 제거할수 있습니다.

2) Kerberos 의 결합

Kerberos 가 비록 일반적이며 엄격한 보안위협을 제거한다고 해도 여러가지 리유로 해서 그것을 실현하기가 어려울것이다:

- 표준 UNIX 통과단어자료기지로부터 /etc/passwd 나 /etc/shadow 나 같은 사용자통과단어들을 Kerberos 통과단어자료기지에로 이동시키는것은 이것을 자동으로 진행하는 미캐니즘이 없기때문에 싫증날수 있습니다.
- Kerberos 는 Pluggable Authentication Modules (PAM) system 와 부분적으로만 호환성을 가집니다. Kerberos 는 매 사용자를 신뢰하지만 신뢰되지 않는 망상에서 신뢰되지 않은 호스트를 리용하고있다고 가정합니다. 그의 기본목적은 그 망을 통하여 전송되고있는 암호화되지 않은 통과단어들을 방지하는것입니다. 그러나 적당한 사용자가 아닌 3 자가 인증에 리용되는 증서를 발급하는 하나의 호스트(key distribution center (KDC)라고 하는)에 대한 접근을 가진다면 전체 Kerberos 인증체계는 위험에 빠집니다.
- Kerberos 를 리용하는 애플리케이션에 대하여 그의 원천은 Kerberos 서고들에 대한 적절한 호출을 진행하도록 변경되여야 합니다. 이 방법으로 변경된 애플리케이션들을 Kerberos-aware, 또는 kerberized 되였다고 말합니다. 일부 애플리케이션들에 대하여 이것은 애플리케이션의 크기나 그의 설계로 인해서 아주 문제시될수 있습니다. 다른 호환되지 않는 애플리케이션들에 대해서는 봉사기와 의뢰기가 통신하는 방법으로 변화되여야 합니다. 다시 이것은 광대한 프로그람작성을 요구할수도 있습니다. 기정으로 Kerberos 지원을 가지지 않는 비공개원천애플리케이션들이 자주 가장 문제성 있습니다.

Kerberos 는 모든것이 아니면 애당초 포기하는 해결책입니다. 만약 망에서 Kerberos 를 리용한다면 비-Kerberos aware 봉사에로 전송되는 암호화되지 않은 통과단어들은 위험합니다. 따라서 망은 Kerberos 의 사용으로부터 아무런 리익도 얻지 못합니다. Kerberos 를 가지고 망을 보안

하기 위해서 암호화되지 않은 통과단어들을 전송하는 모든 client/server 애플리케이션들의 Kerberos-aware 판본들을 리용하든가 그러한 임의의 의뢰기/봉사기애플리케이션들을 전혀 리용하지 말아야 합니다.

2. Kerberos 용어

Kerberos 는 봉사의 여러가지 측면들을 정의하기 위해 자기자체의 용어들을 가지고있습니다. Kerberos 가 어떻게 동작하는가를 배우기전에 다음의 용어들을 배우는것이 중요합니다.

- 인증봉사기 authentication server (AS)

봉사에 대한 접근을 위해서 사용자들에게 차례로 주어지는 요구한 봉사에 대한 증서(ticket)들을 발급하는 봉사기. AS 는 증명서가 없거나 보내지 않은 의뢰기로부터의 요청에 응답합니다. 인증봉사기는 보통 증서허가증서(ticket-granting ticket (TGT))를 발급함으로써 증서허가봉사기(ticket-granting server (TGS))봉사에 대한 접근을 얻는데 리용됩니다. AS 는 보통 열쇠배포중심(key distribution center (KDC))과 같은 호스트상에서 실행합니다.

- 암문 ciphertext

암호화된 자료.

- 의뢰기 client

Kerberos 로 부터 허가증을 접수할수 있는 망에서의 한 부분(여기에는 사용자, 호스트, 혹은 하나의 응용프로그람이 될수 있습니다.)

- 신임장 credentials

특정한 봉사를 위한 의뢰기의 식별자를 검증하는 전자증서입니다.

- 신임장캐쉬 또는 증서화일 credential cache or ticket file

사용자와 여러가지 망봉사들사이에 암호화통신을 위한 열쇠들이 있는 화일을 말합니다. Kerberos 5 는 공유된 기억기와 같은 다른 캐쉬형태사용을 위한 흐레임워크를 지원합니다. 그러나 화일들은 더 많이 지원됩니다.

- 암호해쉬 crypt hash

사용자를 인증하는데 사용되는 한방향하쉬입니다. 이것들은 암호화되

지 않은 자료를 리용하는것보다 더 안전하지만 그것들은 아직 크래커에 의하여 복호화하기 쉽습니다.

- principal (or principal name)

principal 은 Kerberos 를 리용하여 인증이 허용되는 사용자나 봉사의 유일한 이름입니다. Principal 은 형태 root[/instance]@REALM 에 따릅니다. 전형적인 사용자에 대하여 root 는 그의 가입 ID 와 같습니다. Instance 는 선택적입니다. 만약 principal 이 instance 를 가진다면 그것은 root 와 ("/")로 분리됩니다. 빈 문자렬 ("")은 유효한 instance (기정 NULL instance 와 다른)로 여기지만 그것을 리용하면 혼동할수 있습니다. Realm 에서 모든 principal 들은 자기자신의 열쇠를 가지는데 이것은 사용자들에 대하여 통과단어로부터 얻어지거나 봉사들에 대하여 우연적으로 설정됩니다.

- realm

Kerberos 를 사용하는 망이며 KDCs 라고 하는 하나이상의 봉사기들과 잠재적으로 많은수의 의뢰기들로 이루어집니다.

- 봉사 service

망을 통하여 호출되는 프로그람.

- 증서 ticket

특별한 봉사를 위해 의뢰기의 신분을 조사하는 전자신임장들의 림시모임. 신임장이라고도 합니다.

- 증서허가봉사기 ticket-granting server (TGS)

봉사에 대한 접근을 위하여 사용자들에게 차례로 주는 요청봉사에 대한 증서들을 발급하는 봉사기. TGS 는 보통 KDC 와 같은 호스트상에서 실행합니다.

- 증서허가증서 ticket-granting ticket (TGT)

추가적인 증서들을 KDC 로부터 신청하지 않고 의뢰기가 그 증서들을 얻도록 하는 특별한 증서.

3. Kerberos 는 어떻게 동작하는가

Kerberos 는 사용자이름/통과단어 인증방법들과 다르다. 매 망봉사에

대하여 매 사용자를 인증하는 대신에 Kerberos 는 대칭암호화와 신뢰하는 3 자(KDC)를 리용하여 망봉사묶음에 대하여 사용자를 인증합니다. 사용자가 KDC 에 인증할때 KDC 는 사용자의 콤퓨터에로 그 쎄션에 특정한 증서를 되돌려보내며 Kerberos-aware 봉사들은 통과단어를 리용하여 사용자를 인증할것을 요구하는것이 아니라 사용자의 콤퓨터상에서 증서를 찾습니다.

Kerberos-aware 망의 사용자가 자기의 워크스테이션에 가입할때 그의 principal가 인증봉사기로부터 TGT 에 대한 요청부분으로서 KDC 에로 전송됩니다. 이 요청은 그것이 사용자에게 확인되도록 log-in 프로그람에 의해서 보내질수 있거나 사용자가 가입한후에 kinit 프로그람에 의해서 보내질수 있습니다.

그러면 KDC 는 자기의 자료기지에서 그 principal 을 검사합니다. Principal 이 발견되면 KDC 는 TGT 를 작성하는데 이것은 사용자의 열쇠를 리용하여 암호화되며 그 사용자에게로 되돌려집니다.

그 다음 의뢰기상의 Login 이나 kinit 프로그람은 사용자의 열쇠를 리용하여 TGT 를 복호화하는데 이 열쇠는 사용자의 통과단어로부터 계산합니다. 사용자의 열쇠는 오직 의뢰기상에서만 리용되며 망을 통하여 전송되지 않습니다.

TGT 는 일정한 주기의 시간후에 만기로 설정되며 (보통 10 내지 24 시간) 의뢰기의 신임장캐쉬에 보관됩니다. 만기시간은 위태로운 TGT 가 짧은 시간주기동안에만 공격자에게 리용되도록 설정됩니다. TGT 가 발급된후에 사용자는 TGT 가 만기될때까지 혹은 탈퇴하여 다시 가입할때까지 자기의 통과단어를 재입력하지 않아도 됩니다.

사용자가 망봉사에 접근을 요구할때마다 의뢰기쏘프트웨어는 TGT 를 리용하여 TGS 로부터 그 특정한 봉사에 대한 새로운 증서를 요구합니다. 그러면 봉사증서는 그 봉사에 대하여 사용자를 정확하게 인증하는데 리용됩니다.

4. Kerberos 봉사기 구축

Kerberos 를 설치할때 우선 KDC 를 설치해야 합니다. 만약 slave 봉사기들을 설치할 필요가 있다면 master 를 우선 설치합니다.

Kerberos KDC 의 설치단계:

Kerberos 를 구성하기전에 모든 의뢰기들과 봉사기들에서 시간동기화와 DNS 가 정확히 동작하는가를 확인합니다. Kerberos 봉사기와 그의 의뢰기들사이에 시간동기화에 특별한 주목을 돌려야 합니다. 만약 봉사기와 의뢰기사이의 시간차가 5 분이상이라면(이것은 Kerberos 5 에서는 구성가능하다) Kerberos 의뢰기는 봉사기에 인증할수 없습니다. 이 시간동기화는 공격자가 낡은 Kerberos 증서를 리용하여 유효한 사용자로 가장하는것을 방지하기 위해 필요합니다.

지어 Kerberos 를 리용하지 않는다해도 Network Time Protocol (NTP) 호환의 의뢰기/봉사기망을 설정할것을 권고할수 있습니다. 《붉은별》봉사기용체계 3.0 판에는 이 목적을 위한 ntp 패키지가 있습니다.

2. Krb5-libs, krb5-server, krb5-workstation 패키지들을 KDC 를 실행하는 전용콤퓨터에 설치합니다. 이 콤퓨터는 매우 안전해야 할 필요가 있다 — 가능하다면 KDC 외에는 다른 봉사를 실행시키지 말아야 합니다.

3. realm 이름과 령역-realm 넘기기를 반영하는 /etc/krb5.conf 와 /var/kerberos/krb5kdc/kdc.conf 구성화일들을 편집합니다. 간단한 realm 은 EXAMPLE.COM 과 example.com 의 실체들을 정확한 령역이름으로 교체하고 — 대문자이름과 소문자이름들을 정확한 형태로 유지하면서 — KDC 를 kerberos.example.com 으로부터 Kerberos 봉사기의 이름으로 변화시켜 구성할수 있습니다. 규약에 따라 모든 realm 이름들은 대문자이며 모든 DNS 호스트이름들과 령역이름들은 소문자입니다.

4. 쉘프롬프트로부터 kdb5_util 를 리용하여 자료기지를 창조하여야 합니다:

/usr/kerberos/sbin/kdb5_util create -s

create 지령은 Kerberos realm 에 대한 열쇠들을 보관하는 자료기지를 창조합니다. -s 스위치는 주봉사기열쇠가 보관되는 stash 화일의 작성을 지적합니다. 만약 열쇠를 읽을 stash 화일이 존재하지 않는다면 Kerberos 봉사기(krb5kdc) 는 매번 기동할때마다 (열쇠를 재생성하는데

리용될수 있는)주봉사기통과단어를 사용자에게 재촉합니다. /var/kerberos/krb5kdc/kadm5.acl 화일을 편집합니다. 이 화일은 Kerberos 자료기지에 대한 관리접근을 어느 principal 들이 가지는가와 그의 접근준위를 결정하기 위해 kadmind 가 리용합니다. 대부분의 구성은 단일행으로 얻어질수 있다:

*/admin@EXAMPLE.COM *

대부분의 사용자들은 단일 principal (zo@EXAMPLE.COM 와 같이 NULL 이나 빈 instance 를 가진)로 자료기지에 표시됩니다. 이 구성에서 admin 의 인스턴스가 있는 두번째 principal 을 가진 사용자들(실례로 joe/admin@EXAMPLE.COM) 은 그 realm 의 Kerberos 자료기지에 대한 완전한 능력을 장악할수 있습니다.

kadmind 가 봉사기에서 기동한후에 임의의 사용자는 realm 의 임의의 의뢰기들이나 봉사기상에서 kadmin 을 실행함으로써 봉사들에 접근할수 있습니다. 그러나 kadm5.acl 화일에 지정된 사용자들만이 자기의 통과단어들을 변화시키는것을 제외하고 임의의 방법으로 자료기지를 변경시킬수 있습니다.

kadmin 유틸리티는 망을 통하여 kadmind 봉사기와 통신하며 인증을 관리하기 위해 Kerberos 를 리용합니다. 결과적으로 망을 통하여 봉사기와 접속하여 봉사기를 관리하기전에 첫 principal 이 이미 존재해야합니다. kadmin.local 지령으로 첫 principal 을 작성하는데 이것은 특별히 KDC 와 같은 호스트상에서 리용되도록 설계되며 인증에 Kerberos 를 리용하지 않습니다.

첫 principal 을 작성하려면 KDC 터미널에서 다음의 kadmin.local 지령을 입력한다:

/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"

5. 다음의 지령들을 리용하여 Kerberos 를 시작한다:

/sbin/service krb5kdc start

/sbin/service kadmin start

/sbin/service krb524 start

6. kadmin 내의 addprinc 지령을 리용하여 사용자들에 대한 principal 들

을 추가합니다. kadmin 과 kadmin.local 은 KDC 에 대한 지령행결합부입니다. kadmin 프로그람을 시작한후에 addprinc 와 같은 많은 지령들을 리용할수 있습니다.

7. KDC 가 증서들을 발급하는가를 조사합니다. 우선 kinit 를 실행하여 증서를 얻어 그것을 신임장캐쉬에 보관합니다. 다음으로 klist 를 리용하여 캐쉬내의 신임장목록을 보고 kdestroy 를 리용하여 캐쉬와 그 안에 있는 신임장들을 파괴합니다.

주의

기정으로서 kinit 는 같은 체계의 login 사용자이름(Kerberos 봉사기가 아니라)을 리용하여 인증을 시도합니다. 만약 그 사용자이름이 Kerberos 자료기지의 principal 에 대응되지 않는다면 kinit 는 오유통보문을 내보낸다. 그렇게 된다면 지령행인수로서 정확한 principal 이름을 주어 kinit 를 실행시켜야 합니다(kinit <principal>).

일단 이 단계들이 완성된다면 Kerberos 봉사기는 실행되고있는것입니다.

5. Kerberos 5 의뢰기 구축

의뢰기의 설정은 봉사기설정보다 적은 공정을 포함합니다. 의뢰기 패키쥐를 설치하고 krb5.conf 구성화일을 설정하여야 합니다. ssh 와 slogin 이 의뢰기 체계에서 원격으로 가입하는 방법이라면 kerberized 된 rsh 와 rlogin 은 몇가지 구성에서의 변화를 요구합니다.

1. Kerberos 의뢰기와 KDC 사이의 시간동기화에 주의하여야 합니다. 추가적으로 Kerberos 의뢰기를 구성하기 전에 의뢰기에서 DNS 가 정확히 동작하는가를 확인하여야 합니다.
2. Krb5-libs 와 krb5-workstation 패키쥐를 모든 의뢰기 콤퓨터들에 설치하여야 합니다. 다음 /etc/krb5.conf 화일을 설정하여야 합니다. (보통 이것은 KDC 의 krb5.conf 화일과 같습니다.)
3. Ssh 나 Kerberized 된 rsh, rlogin 을 리용하여 련결하는 사용자들을

인증하기 위하여 령역에서 Kerberos 를 리용하는데 그 전에 Kerberos 자료기지에서 자기의 호스트 주실체를 가지고있어야 합니다. Sshd, kshd, klogind 봉사기 프로그람들은 모두 호스트 봉사의 주실체를 위한 열쇠들을 리용한 접근이 필요합니다. 추가적으로 rsh, rlogin 봉사를 리용하기 위하여서는 xinetd 패키쥐를 설치하여야 합니다.

Kadmin 을 리용하여 KDC 에서 workstation 을 위한 호스트 주실체를 추가하여야 합니다. 이 경우 workstation 의 호스트이름입니다. 다음 –randkey 항목을 리용하여 kadmin 의 addprinc 지령을 실행함으로서 임의의 열쇠를 가진 주실체를 생성할수 있습니다.

addprinc –randkey host/blah.example.com

다음 주실체가 생성되면 ktadd 지령을 리용하여 열쇠를 등록합니다.

ktadd –k /etc/krb5.keytab host/blah.example.com

제 7 절. 가상사설망

실례로 많은 기업체들은 비동기전송방식을 리용하고있습니다. 이것은 서로 끝과 끝을 이은 망구조입니다. 이렇게 하면 가격이 매우 높아지게 되며 특히 중소규모 기업들에는 큰 문제로 됩니다.

이러한 문제를 해결하기 위하여 VPN 이 개발되였습니다. VPN 은 두부분사이의 보안된 수자통신을 진행하도록 하며 국부대역망으로부터 광대역망을 생성합니다.

- VPN 은 어떻게 동작하는가?

파케트가 의뢰기로부터 전송되면 경로와 인증을 위한 인증머리부를 추가한 VPN 경로기나 관문을 통하여 전송됩니다. 자료는 암호화된 다음 Encapsulating Security Payload(ESP) 에 포함됩니다.

접수한 VPN 경로기는 머리부정보를 자른다음 그 자료를 복호화하고 목적지에 그것을 전송합니다. 망과 망사이의 련결을 사용할때 국부망에서 접수한 마디는 이미 복호화된 파케트를 접수하며 처리를 준비합니다.

보안의 준위가 높으면 공격자는 파케트를 가로챈다고 하여도 그것을

복호화할수는 없습니다.

제 8 절 방화벽

방화벽들은 망보안실현물의 핵심적인 부분품들중의 하나입니다. 방화벽들은 Cisco 와 Nokia, Sonicwall 이 제공하는 방화벽기구들과 같이 단독가동방식의 하드웨어해결책으로 될수 있습니다. Checkpoint 와 McAfee, Symantec 와 같은 쏘프트웨어방화벽해결책들을 개발하였습니다.

하드웨어와 쏘프트웨어방화벽들사이에 차이점이 있는것외에도 하나의 해결책과 다른 해결책을 분리하는 방화벽기능에서도 차이점이 존재합니다. 표.《방화벽형태》에서는 3 가지의 일반적인 형태의 방화벽들과 그 기능들에 대하여 상세하게 보여줍니다.

표. 방화벽형태

방법	설명	우점	결함
NAT	망주소변환은 공개 IP 주소들중 하나 혹은 작은 구역안에 있는 사설 IP 부분망들에 놓이며 여러개의 원천이 아니라 하나의 원천에 대한 모든 요청들을 조작(masquerade)합니다. Linux 핵심부는 망려과핵심부 부분체계를 통하여 내부적으로 구축된 NAT 기능을 포함하고있습니다.	* LAN 상의 콤퓨터들에 개방적으로 설정될수 있습니다. * 하나이상의 외부 IP 주소들뒤에 있는 많은 콤퓨터들과 봉사들의 보호는 관리업무를 간단하게 합니다. * LAN 에 대한 사용자접근의 제한은 NAT 방화벽 및 관문에서 포구를 열기,	* 방화벽밖에 있는 봉사에 접속하면 악성활동을 막을수 없습니다.

		닫기함으로써 설정될수 있습니다.	
파케트려과기능	파케트려과기능을 가진 방화벽은 LAN 을 통하여 통신하는 매 자료파케트들을 읽습니다. 그것은 머리부정보에 의하여 파케트들을 읽고 처리할수 있으며 방화벽관리자에 의하여 실현된 프로그람작성가능한 규칙모임에 기초하여 파케트들을 려과합니다. Linux 핵심부는 망려과핵심부부분체계를 통하여 내부적으로 구축되여있는 파케트려과기능을 포함하고있습니다.	* iptables front-end 프로그람을 통하여 개작할수 있습니다. * 의뢰기측에 대한 개작은 필요로 하지 않습니다. 왜냐면 모든 망활동이 응용프로그람준위가 아니라 경로기준위에서 려과되기때문입니다. * 파케트들이 대리봉사기를 통하여 전송되지 않기때문에 망성능이 의뢰기와 원격호스트사이 직접접속으로 인하여 보다 더 향상됩니다.	* 대리봉사기방화벽들과 같은 내용에 대한 파케트들을 려과할수 없습니다. * 규약층에 있는 파케트들을 처리하지만 응용프로그람층에 있는 파케트들은 려과할수 없습니다. * 복잡한 망구성방식은 파케트려과규칙확립을 어렵게 할수 있습니다. 특히 IP 꾸밈이나 국부적인 부분망과 DMZ 망이 결합되는 경우 어렵게 됩니다.
대리봉사기	대리봉사기방화벽들은 LAN 의뢰기에서 대리봉사	* 관리자들에게 LAN 밖에서 동작하는 응용	* 대리봉사기들은 응용프로그람(HTTP,

	기콤퓨터에로 가는 어떤 규약이나 형태들의 모든 요청들을 려과하며 그 다음 국부적인 의뢰기를 대신하여 인터네트에 대한 요청들을 작성합니다. 대리봉사기콤퓨터는 악의있는 원격사용자들과 내부망의뢰기콤퓨터들사이의 완충기로서 동작합니다.	프로그람들과 규약들에 대한 조종권한을 줍니다. * 일부대리봉사기봉사기들은 인터네트봉사를 리용하는것이 아니라 자주 접근된 자료들을 국부적으로 캐쉬할수 있습니다. 이것은 대역(bandwidth)의 소모를 줄일수 있습니다. * 대리봉사기봉사들은 엄밀하게 기록되고 감시되며 망상에 있는 자원리용에 대한 조종을 강화하게 합니다.	Telnet 등)에 따라 다르거나 규약제한(대부분의 대리봉사기들은 TCP 접속봉사만으로 동작한다)을 받습니다. * 응용프로그람봉사들은 대리봉사기뒤에서 실행할수 없으며 따라서 응용프로그람봉사기들은 분리되여있는 망보안형태를 리용해야 합니다. * 대리봉사기들은 좁은 망통로(network bottleneck)로 될수 있습니다. 그것은 모든 요청들과 전송이 의뢰기에서 원격봉사에로의 직접적인것이 아니라 하나의 원천을 통하여 처리되기때문입니다.

1. Netfilter 과 IPTables

Linux 핵심부는 Netfilter 라는 강력한 망관리부분체계를 포함하고있습니다. 망려과부분체계는 NAT 와 IP 꾸밈봉사들뿐 아니라 모든 (stateful and stateless)파케트려과기능을 제공합니다. 망려과기는 또한 고급한 경로기능과 접속상태관리를 위한 IP 머리부정보를 조작(mangle)할수 있는 능력도 가지고있습니다. 망려과기는 iptables 도구를 리용하여 조종됩니다.

망려과기의 능력과 유연성(flexibility)은 iptables 관리도구를 리용하여 실현됩니다. Linux 핵심부 2.4 이전판본에서는 iptables 와 류사한 도구인 ipchains 를 리용하였습니다.

Iptables 는 망접속과 망점검, 망처리를 강화하기 위하여 망려과부분체계를 리용하고있습니다. Iptables 는 고급한 로그관리, 전경로조종(pre-routing)과 후경로조종(post-routing), 망주소변환, 포구회송(forwarding), 하나의 지령행대면부안에서의 모든 동작들로 특징지어집니다.

이 절에서는 iptables 의 개념을 제공합니다.

2. 기초방화벽설정

건물안의 불막이벽(firewall)이 불이 퍼지지 않도록 하는것과 마찬가지로 콤퓨터방화벽은 악성쏘프트웨어가 콤퓨터에 전파되지 않도록 합니다. 그것은 또한 허가되지 않은 사용자들이 콤퓨터에 접근하지 못하도록 합니다.

《붉은별》봉사기용체계기정설치에서 방화벽은 사용자의 콤퓨터나 망과 어떤 신뢰되지 않는 망들, 실례로 인터네트사이에 존재합니다. 그것은 원격사용자들이 콤퓨터상의 어느 봉사들에 접근할수 있는가를 결정합니다. 정확하게 설정된 방화벽은 체계보안을 상당히 개선할수 있습니다. 사용자는 인터네트에 접속하고있는 임의의 《붉은별》봉사기용체계 3.0 판체계에 대하여 방화벽을 설정하여야 합니다.

1) 방화벽의 사용가능 및 사용불가능설정

방화벽에 대하여 다음과 같은 선택항목들중 하나를 선택합니다.

- Disabled(사용불가능) – 방화벽사용불가능은 사용자의 체계에 완전히 접근하도록 하며 보안검사를 진행하지 않습니다. 이것은 사용자가 신뢰되는(인터네트가 아닌) 망상에서 체계를 실행하고있는 경우에만 설정되여야 하며 혹은 iptables 지령행도구를 리용하여 전용방화벽을 설정하는데 필요하게 됩니다.

  경고 : 방화벽설정과 임의의 전용화된 방화벽규칙들은 /etc/sysconfig/iptables 화일에 보관됩니다. 만일 사용자가 Disabled 를 선택하고 OK 를 누르면 이 설정과 방화벽규칙들이 삭제되게 됩니다.

- Enabled(사용가능) – 이 항목은 체계가 DNS 응답이나 DHCP 요청과 같이 외부로 나가는 요청들에 응답하지 않는 들어오는 접속들을 거부하도록 설정합니다. 만일 이 콤퓨터에서 실행하고있는 봉사들에 대한 접근이 필요한 경우 사용자는 방화벽을 통하여 특정의 봉사들을 선택하여 허가할수 있습니다.

  만일 사용자가 체계를 인터네트에 접속하려고 하지만 봉사기를 실행하려고 계획하고있지 않는 경우 이것은 가장 안전한 선택으로 됩니다.

2) 신뢰하는 봉사들

Trusted Services 목록안의 선택항목들을 사용할수 있게 설정하면 특정의 봉사들이 방화벽을 통과하도록 허가합니다.

WWW(HTTP)

HTTP 규약은 Apache(기타 웨브봉사기)에 의하여 웨브페지들을 봉사하는데 리용됩니다. 만일 사용자가 웨브봉사기를 공개적으로 사용할수 있도록 설정하려고 계획하고있다면 이 검사칸(check box)을 설정하여야 합니다. 이 항목은 웨브페지를 개발하거나 국부적으로 페지들을 조사하는데서는 요구되지 않습니다. 이 봉사는 httpd 패키지가

설치될것을 요구합니다.

WWW(HTTP)을 사용할수 있도록 설정한다고 해도 HTTPS, HTTP 의 SSL 판본에 대한 포구는 열리지 않습니다. 만일 이 봉사기 필요하게 되는 경우 Secure WWW(HTTPS)검사칸을 선택하여야 합니다.

FTP

FTP 규약은 망상의 콤퓨터들사이에 화일들을 전송하는데 리용됩니다. 만일 사용자가 FTP 봉사기를 공개적으로 사용할수 있도록 설정하려고 계획하고있다면 이 검사칸을 선택하여야 합니다. 이 봉사는 vsftpd 패키지가 설치되여있을것을 요구합니다.

SSH

보안쉘(Secure Shell)은 원격콤퓨터에 가입하여 지령을 실행하기 위한 지령묶음입니다. ssh 를 통하여 콤퓨터에 대한 원격접근을 허가하려면 이 검사칸을 선택하여야 합니다. 이 봉사는 openssh-server 패키지가 설치되여있을것을 요구합니다.

Telnet

Telnet 는 원격콤퓨터에 가입하기 위한 규약입니다. Telnet 통신들은 암호화되지 않으며 망조사에 대하여 그 어떤 보안도 제공하지 않습니다. 내부로 들어오는 Telnet 접근에 대한 허가는 권고하지 않습니다. telnet 를 통한 콤퓨터의 원격접근을 허가하려면 이 검사칸을 선택하여야 합니다. 이 봉사는 telnet-server 패키지가 설치되여있을것을 요구합니다.

Mail(SMTP)

SMTP 는 원격콤퓨터가 사용자의 콤퓨터에 우편을 발송하기 위하여 직접 접근하도록 하는 규약입니다. 만일 사용자가 ISP 의 봉사기로부터 POP3이나 IMAP를 리용하여 우편을 수집하거나 fetchmail과 같은 도구를 리용한다면 이 봉사를 사용하지 말아야 합니다. 사용자의 콤퓨터에 우편발송을 허가하려면 이 검사칸을 선택하여야 합니다. 부정확하게 설정된 SMTP 봉사기는 원격콤퓨터가 spam 을 전송하기 위하여 봉사기를 리용하도록 허가할수 있다는것을 주의해야 합니다.

NFS4

망화일체계(NFS)는 *NIX 체계들에서 공통적으로 리용되는 화일공유 규약입니다. 이 규약의 판본 4 는 이전 판본들에 비해볼 때 보다 더

안전합니다. 만일 사용자가 다른 망사용자들과 자기의 체계에서 화일이나 등록부들을 공유하려고 한다면 이 검사칸을 선택하여야 합니다.

Samba

Samba는 Microsoft의 독점적인 SMB망관리규약의 실현물입니다. 만일 사용자가 Microsoft Windows 콤퓨터들을 가지고 화일이나 등록부들, 국부적으로 련결된 인쇄기들을 공유하여야 한다면 이 검사칸을 선택하여야 합니다.

3) 기타포구들

방화벽설정도구는 전용 IP 포구들을 iptables 에 의해 신뢰되는것으로 정의하기 위한 Other ports 부분을 포함하고있습니다. 실례로 IRC 와 인터네트인쇄관리규약(IPP-Internet printing protocol)이 방화벽을 통과하도록 하려면 다음의 행을 Other Ports 부분에 추가해야 합니다.

4) 설정보관

OK 를 눌러 변경된 내용들을 보관하고 방화벽을 사용가능 혹은 사용불가능으로 설정합니다. 만일 Enable firewall 이 선택되였다면 선택된 항목들은 iptables지령들에 옮겨지며 /etc/sysconfig/iptables화일에 작성됩니다. Iptables 봉사는 또한 방화벽이 설정된 선택항목들을 보관한 다음 즉시 능동으로 되도록 기동되여있습니다. 만일 Disable firewall 이 선택되였다면 /etc/sysconfig/iptables 화일이 삭제되며 iptables 봉사는 즉시 정지됩니다.

또한 선택된 항목들은 설정들이 응용프로그람이 다음번에 기동될 때 회복될수 있도록 하기 위하여 /etc/sysconfig/system-config-securitylevel 화일에 기록됩니다. 이 화일을 수동적으로 편집하지는 말아야 합니다.

방화벽이 즉시 능동으로 된다고 해도 iptables봉사는 기동시에 자동적으로 기동하도록 설정되여있지 않습니다.

5) IPTables 봉사활성화

방화벽규칙들은 오직 iptables봉사가 실행되고있는 경우에만 능동으로 됩니다. 봉사를 수동적으로 기동하려면 다음의 지령을 리용하여야 합니다.

[root@myServer~]#service iptables restart

체계가 기동될 때 iptables 가 기동하도록 하려면 다음의 지령을 리용하여야 합니다.

[root@myServer ~]#chkconfig –level 345 iptables on

3. IPTables 리용방법

Iptables 리용에서 첫 단계는 iptables 봉사를 기동하는것입니다. 다음의 지령을 리용하여 iptables 봉사를 기동합니다.

[root@myServer~]#service iptables start

주의 : ip6tables 봉사는 만일 사용자가 iptables 봉사만을 리용하려고 계획하고있는 경우는 중지될수 있습니다. 만일 사용자가 ip6tables 봉사를 비활성화시킨다면 IPv6 망도 비활성화시켜야 한다는것을 잊지 말아야 합니다. 정합하는 방화벽이 없는 망장치는 능동으로 설정하지 말아야 합니다.

Iptables 를 체계기동시에 기정으로 기동하도록 하려면 다음의 지령을 리용하여야 합니다.

[root@myServer~]#chkconfig –level 345 iptables on

이것은 iptables 가 체계가 실행준위 3 이나 4, 5 로 기동될 때마다 기동하도록 합니다.

1) IPTables 지령문맥

다음의 iptables 지령실례는 기초적인 지령문법을 보여주고있습니다.

[root@myServer~]#iptables –A <chain> -j <target>

-A 선택항목은 규칙이 <chain>(사슬)에 추가된다는것을 정의합니다. 매 사슬은 하나이상의 규칙들로 구성되며 따라서 규칙모임이라고도 합니다.

내부적으로 구축된 3개의 사슬들은 INPUT와 OUTPUT, FORWARD입니다. 이 사슬들은 영구적이며 삭제될수 없습니다. 사슬은 파케트가 조종되는 지점을 정의합니다.

-j <target>선택항목은 규칙의 목표(target)를 정의합니다. 즉 만일 파케트가 규칙과 일치하면 무엇을 해야하는가를 정의합니다. 내부적으로 구축된 목표들은 ACCEPT 와 DROP, REJECT 입니다.

리용가능한 사슬들과 선택항목들, 목표들에 대한 보다 더 상세한 정보는 iptables 도움말을 참고하여 알수 있습니다.

2) 기초방화벽방책

기초적인 방화벽방책의 확립은 보다 더 상세하고 사용자정의의 규칙들을 구축하기 위한 기초를 창조합니다.

매 iptables 사슬은 기정방책과 방화벽의 전반적인 규칙모임을 정의하기 위하여 기정방책과 협력하여 동작하는 하나이상의 규칙들로 구성되여있습니다.

사슬에 대한 기정방책은 DROP 나 ACCEPT 일수 있습니다. 보안에 관심을 가지는 관리자들은 일반적으로 DROP 의 기정방책을 실현하며 경우에 따라 특정의 파케트들만을 허가합니다. 실례로 다음의 방책들은 망관문에서 들어오고 나가는 파케트들을 모두 차단합니다.

```
[root@myServer~]#iptables  –P  INPUT  DROP
[root@myServer~]#iptables  –P  OUTPUT  DROP
```

임의의 회송된 파케트들, 즉 방화벽으로부터 그의 목적마디까지 경로조종되는 망통화량이 거부되여야 하며 내부의뢰기들이 부주의로 인하여 Internet 에로 로출되지 않도록 제한합니다. 이것을 다음의 규칙을 리용하여 진행합니다.

```
[root@myServer~]#iptables  –P  FORWARD  DROP
```

사용자가 매 사슬에 대하여 기정방책을 설정하였을 때 사용자는 특수한 망과 보안요구조건들에 대한 규칙들을 창조하고 보관할수 있습니다.

다음절들에서는 iptables 규칙보관방법과 iptables 방화벽을 구축하는 과정에 실현할수 있는 일부 규칙들에 대하여 설명합니다.

3) IPTables 규칙보관과 회복

Iptables 에 대한 변경은 일시적입니다. 만일 체계가 재기동하거나 iptables 봉사가 재기동한다면 규칙들은 자동적으로 flush 되고 재설정됩

니다. Iptables 봉사가 기동될 때 규칙들이 자동적으로 설정되도록 하려면 다음의 지령을 리용하여야 합니다.

[root@myServer~]#service iptables save

규칙들은 /etc/sysconfig/iptables 화일에 보관되며 봉사가 기동되거나 콤퓨터가 재기동될 때마다 적용됩니다.

4. 일반적인 IPTables 려과기능

원격공격자들이 LAN 에 접근하지 못하도록 하는것은 망보안에서 가장 중요한 문제들중의 하나입니다. LAN 의 완전성은 엄격한 방화벽규칙들을 리용하여 악의있는 원격사용자들로부터 보호되여야 합니다.

그러나 들어오고 나가며 회송되는 파케트들을 모두 차단하도록 설정한 기정방책을 가지고는 방화벽 및 관문과 내부 LAN 사용자들이 서로 혹은 외부자원들과 통신하도록 하는것은 불가능합니다.

사용자들이 망관련기능들을 집행하도록 하며 망관리응용프로그람들을 리용하도록 하기 위하여 관리자들은 통신을 위한 포구들을 열어놓아야 합니다.

실례로 방화벽에서 80 포구에 대한 접근을 허가하려면 다음의 규칙을 추가해야 합니다.

[root@myServer~]#iptables –A INPUT –p tcp –m tcp –dport 80 –j ACCEPT

이것은 사용자들이 표준포구 80 을 리용하여 통신하는 웨브싸이트들을 열람하도록 허가합니다. 안전한 웨브싸이트들(실례로 https://www.example.com/)에 접근하도록 허가하려면 다음과 같이 포구 443 에 대한 접근을 제공하여야 합니다.

[root@myServer~]#iptables –A INPUT –p tcp –m tcp –dport 443 –j ACCEPT

중요:

Iptables 규칙모임을 창조할 때 그 순서가 중요합니다.

만일 어떤 규칙이 192.168.100.0/24 부분망으로부터 들어오는 임의의 파케트들을 차단(drop)한다는것을 정의하고 이것이 192.168.100.13(삭제

되는 부분망안에 있음)으로부터 들어오는 파케트들을 허가하는 규칙다음에 놓인다면 두번째 규칙은 무시됩니다.

192.168.100.13 으로부터 들어오는 파케트들을 허가하는 규칙은 부문망의 나머지 주소들을 차단하는 규칙들이 앞에 놓여야 합니다.

현존사슬안의 특정의 위치에 어떤 규칙을 삽입하려면 –I 선택항목을 리용하여야 합니다. 실례로

[root@myServer~]#iptables –I INPUT 1 –i lo –p all –j ACCEPT

이 규칙은 국부 loopback 장치통화를 허가하기 위하여 INPUT 사슬안에 첫번째 규칙으로서 삽입됩니다.

사용자가 LAN 에로의 원격접근을 요구는 여러번 있을수 있습니다. 안전한 봉사들, 실례로 SSH 는 LAN 봉사들에 대한 암호화된 원격접속에 리용될수 있습니다.

PPP 기초의 자원들(모뎀이나 ISP 계정들과 같은), dial-up 접근을 가지고 있는 관리자들은 방화벽장벽들을 안전하게 우회하는데 리용될수 있습니다. 그것들은 직접적인 접속이기때문에 모뎀접속들은 일반적으로 방화벽 및 관문뒤에 놓입니다.

그러나 광대역접속을 가진 원격사용자들에 대하여서는 특별한 경우가 있을수 있습니다. 사용자는 iptables 를 설정하여 원격 SSH 의뢰기들로부터 접속들을 접수하도록 할수 있습니다. 실례로 다음의 규칙들은 원격 SSH 접근을 허가합니다.

[root@myServer~]#iptables –A INPUT –p tcp –dport 22 –j ACCEPT

[root@myServer~]#iptables –A OUTPUT –p tcp –sport 22 –j ACCEPT

이 규칙들은 인터네트나 방화벽 및 관문에 직접 접속된 단일 PC 와 같이 개별적인 체계에 대하여 들어오고 나가는 접근을 허가합니다. 그러나 그것들은 방화벽 및 관문뒤에 있는 마디들이 이 봉사들에 접근하는것을 허가하지는 않습니다. 이 봉사들에 대한 LAN 접근을 허가하기 위하여 사용자는 iptables 려과규칙들을 가지고 망주소변환(NAT)을 리용할수 있습니다.

5. FORWARD 와 NAT 규칙들

대부분의 ISP 들은 공개적으로 경로조종가능한 제한된 IP 주소들만을 그것들이 봉사하는 조직들에 제공합니다.

따라서 관리자는 LAN 상의 모든 마디에 공개 IP 주소를 제공하지 않고 인터네트봉사들에 대한 접근을 공유하기 위한 또 다른 방법들을 발견하여야 합니다. 사설 IP 주소들의 리용은 LAN 상의 모든 마디들이 내부와 외부망봉사들에 정확히 접근하도록 허가하는 가장 일반적인 방법입니다.

방화벽들과 같은 경로기들은 인터네트로부터 들어오는 전송자료들을 받을수 있으며 그 파케트들을 계획된 LAN 마디들에 경로조종할수 있습니다. 동시에 방화벽 및 관문들은 또한 LAN 마디로부터 원격인터네트봉사들에로 나가는 요청들도 경로조종할수 있습니다.

이 망통화회송은 때때로 위험하게 될수 있으며 특히는 고급한 크랙킹도구들을 리용하면 내부 IP 주소들을 위장할수 있고 원격공격자의 콤퓨터를 사용자의 LAN 상의 마디로써 동작하도록 할수 있습니다.

이것을 막기 위하여 iptables 는 망자원들에 대한 비정상적인 사용을 막도록 구현될수 있는 경로조종과 회송방책들을 제공합니다.

FORWARD 사슬은 파케트들이 LAN 안에 경로조종될수 있도록 관리자들의 조종을 허가합니다. 전체적인 LAN 에 대한 회송을 허가하려면 (방화벽 및 관문이 eth1 에서 내부 IP 주소를 할당받는다고 가정) 다음의 규칙들을 리용해야 합니다.

```
[root@myServer~]#iptables –A FORWARD –i eth1 –j ACCEPT
[root@myServer~]#iptables –A FORWARD –o eth1 –j ACCEPT
```

이 규칙은 방화벽 및 관문뒤에 있는 체계들에 내부망에 대한 접근을 제공합니다. 관문은 파케트들이 LAN 마디에서 자기의 예정된 목적마디로 가도록 경로를 조종하며 자기의 eth1 장치를 통하여 모든 파케트들을 통과합니다.

주의:

기정으로 《붉은별》봉사기용체계 3.0 판핵심부들안에 있는 IPv4 방책은 IP 회송에 대한 지원을 사용할수 없습니다. 이것은 봉사기용체계

가 가동하는 콤퓨터들이 경로기로서 동작하지 못하도록 합니다. IP 회송을 사용할수 있게 하려면 다음의 지령을 리용하여야 합니다.

[root@myServer~]#sysctl –w net.ipv4.ip_forward=1

이 설정변경은 현재 쎄션에 대하여서만 유효합니다. 그것은 재기동이나 망봉사재기동후에는 존재하지 않습니다. IP 회송설정을 영구적으로 보존하려면 다음과 같은 /etc/sysctl.conf 화일을 편집하여야 합니다.

다음과 같은 행이 존재합니다.

net.ipv4.ip_forward = 0

다음과 같이 편집합니다.

net.ipv4.ip_forward = 1

우의 변경을 사용할수 있게 하려면 다음의 지령을 리용하여야 합니다.

[root@myServer~]#sysctl –p /etc/sysctl.conf

1) Postrouting 과 IP 꾸밈

회송된 파케트들이 방화벽의 내부 IP 장치를 통하여 접수하는것은 LAN 마디들이 서로 통신하도록 허가합니다. 그러나 그것들은 여전히 인터네트와 외부적으로 통신할수 없습니다.

사설 IP 주소들을 가지고있는 LAN 마디들이 외부공개망들과 통신하도록 허가하기 위하여 IP 꾸밈에 대한 방화벽을 설정하여야 합니다. 그것은 방화벽의 외부장치의 IP 주소를 가지고있는 LAN 마디들에서 오는 요청들을 마스크합니다.(이 경우에는 eth0)

[root@myServer~]#iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

이 규칙은 NAT 파케트정합표(-t nat)를 리용하여 방화벽의 외부망관리장치(-o eth0)에서 NAT(-A POSTROUTING)에 대한 내부적으로 구축되여있는 POSTROUTING 사슬을 정의합니다.

POSTROUTING 은 파케트들이 방화벽의 외부장치를 떠나기때문에 변경되도록 허가합니다.

-j MASQUERADE 목적은 방화벽 및 관문의 외부 IP 주소를 가진 마디의 사설 IP 주소를 마스크하기 위하여 정의되여있습니다.

2) Prerouting

만일 사용자가 외부에서 사용할수 있도록 하는 봉사기를 내부망에 가지고있다면 사용자는 PTREROUTING 사슬의 –j DNAT 를 리용하여 사용자의 내부봉사에 대한 접속을 요청하는 내부로 들어오는 파케트들이 회송될수 있는 목적 IP 주소와 포구를 정의합니다.

실례로 사용자가 들어오는 HTTP 요청들을 172.31.0.23 의 Apache HTTP 봉사기에 회송하려고 한다면 다음의 지령을 리용하여야 합니다.

[root@myServer ~] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 172.31.0.23:80

이 규칙은 nat 표가 내부적으로 구축된 PREROUTING 사슬을 리용하여 목록화된 172.31.0.23 의 목적 IP 주소에 독점적으로 들어오는 HTTP 요청들을 회송하도록 합니다.

주의:

만일 사용자가 FORWARD 사슬안에 DROP 의 기정방책을 가지고있다면 들어오는 모든 HTTP 요청들을 회송하는 규칙을 추가하여 목적 NAT 경로조종이 가능할수 있도록 하여야 합니다. 이것을 집행하려면 다음의 지령을 리용하여야 합니다.

[root@myServer ~] # iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 172.31.0.23 -j ACCEPT

이 규칙은 방화벽으로부터 들어오는 모든 HTTP 요청들을 예정된 목적지로 회송합니다. Apache HTTP 봉사기는 방화벽뒤에 놓입니다.

3) DMZs 와 IPTables

사용자는 iptabels 규칙을 창조하여 HTTP 나 FTP 봉사기와 같이 《비무장지역(DMZ-demilitarized zone)》에 있는 어떤 콤퓨터들에 대한 통화를 경로조종할수 있습니다. DMZ 는 인터네트와 같이 공개된 곳(carrier)에서 봉사를 제공하도록 되여있는 특별한 국부부분망입니다.

실례로 10.0.4.2 에 선정된 HTTP 봉사기에 대하여 들어오는 HTTP 요청(LAN 의 192.168.1.0/24 의 외부에서)들을 경로조종하기 위한 규칙을 설정하기 위하여 NAT는 PREROUTING표를 리용합니다. NAT는 이 표를 리용하여 파케트들을 적당한 목적지로 회송합니다.

[root@myServer ~] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT –to -destination 10.0.4.2:80

이 문서에서 LAN 의 외부에서 80 포구에로의 모든 HTTP 접속들은 내부망으로부터 분리되여있는 망상에 있는 HTTP 봉사기로 경로조종됩니다. 이와 같은 형태의 망토막자료들은 망상의 콤퓨터상에 있는 HTTP 접속들을 허가하는것보다 더 안전하게 립증할수 있습니다.

만일 HTTP 봉사기가 보안접속을 받아들이도록 설정되여있다면 443 포구가 회송되여야 합니다.

6. 악성쏘프트웨어와 위장된 IP 주소

보다 더 섭세한 규칙들이 LAN 안에서 특수한 부분망이나 특수한 마디들에 대한 접근을 조종하도록 창조될수 있습니다. 사용자는 트로이목마와 웜, 기타 의뢰기 및 봉사기비루스들과 같은 어떤 명백치 않은 응용프로그람들이나 프로그람들이 봉사기에 접촉하지 못하도록 제한할수도 있습니다.

실례로 일부 트로이들은 31337 로부터 31340 까지의 포구에서의 봉사를 위한 망들을 검색한다(크랙킹용어에서 elite 포구라고 함).

이와 같은 비표준포구들을 통하여 통신하는 합법적인 봉사들은 존재하지 않기 때문에 그것들에 대한 차단은 사용자의 망상에 있는 감염된 마디들이 개별적으로 자기의 원격주봉사기와 통신할수 있는 기회를 줄일수 있습니다.

다음의 규칙들은 포구 31337 을 리용하려고 하는 모든 TCP 통화를 차단합니다.

[root@myServer ~] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP

[root@myServer ~] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP

사용자는 또한 LAN 에 침투하기 위하여 사설 IP 주소범위를 속여넘기려고 하는 외부접속들을 차단할수도 있습니다.

실례로 사용자의 LAN 이 192.168.1.0/24 를 리용하고있다면 사용자는 인터네트로 향한(internet-facing) 망장치(실례로 eth0)가 사용자의 LAN

IP 범위안에 있는 주소를 가진 장치에 대한 임의의 파케트들을 차단하도록 하는 규칙을 설계할수 있습니다.

기정방책에서는 회송된 파케트들을 거부하도록 되여있기때문에 외부로 향한 장치(eth0)에 대한 기타 위장된 IP 주소는 자동적으로 거부됩니다.

[root@myServer ~] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP

주의:

추가된 규칙들을 처리할 때 DROP 와 REJECT 목표사이에는 차이가 있습니다.

REJECT 목표는 접근을 거부하고 봉사에 접속하려고 하는 사용자들에게 접속거부(connection refused)오유를 되돌립니다. DROP 목표는 이름이 나타내는것과 같이 그 어떤 경고도 출력하지 않고 파케트들을 차단합니다.

관리자들은 목적에 따라 임의의 목표들을 리용할수 있습니다. 그러나 사용자혼잡을 피하고 련속적인 접속을 허가하려고 하는 경우에는 REJECT 목표를 리용할것을 권고합니다.

7. IPTables 와 접속추적방법(tracking)

사용자는 접속상태에 기초하여 봉사들에 대한 접속을 조사하고 제한할수 있습니다. Iptables 안의 모듈은 들어오는 접속들에 대한 정보를 보관하기 위하여 접속추적관리(connection tracking)라고 하는 방법을 리용합니다. 사용자는 다음과 같은 접속상태들에 기초하여 접근을 허가하거나 거부할수 있습니다.

- NEW – HTTP 요청과 같이 새로운 접속을 요청하는 파케트.
- ESTABLISHED – 현존접속에 속하는 파케트.
- RELATED – 새로운 접속을 요청하지만 현존접속에 속하는 파케트. 실례로 FTP 는 21 포구를 리용하여 접속을 확립하지만 자료는 다른 포구(표준적으로는 20 포구)를 리용하여 전송됩니다.

- INVALID – 접속추적관리표안의 접속들에 속하지 않는 파케트.

사용자는 규약이 stateless(UDP 와 같은)이라고 해도 임의의 망규약을 가지고 iptables 접속추적관리의 stateful 기능을 리용할수 있습니다. 다음의 실례에서는 접속추적관리기능을 리용하여 확립된 접속과 련관되여 있는 파케트들만을 회송하는 규칙을 보여줍니다.

[root@myServer ~] # iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

제 9 절 IPTables

봉사기용체계들에는 망파케트려과를 진행하는 많은 도구들이 포함되여있습니다. 핵심부 판본 2.4 이전에는 파케트를 려과하기 위한 ipchains 에 의존하여 려과공정의 매 단계에 파케트들에 대한 규칙목록을 리용하였습니다. 2.4 에서 iptables 가 소개되였습니다.

이 장에서는 iptables 지령들과 려과규칙들이 체계재기동시 어떻게 실행되는가를 보여주며 파케트려과에 초점을 둡니다.

1. 파케트 려과

봉사기핵심부는 Netfilter 를 사용하여 파케트를 려과함으로써 통과허용된 일부만을 접수하고 다른것들은 거부하게 할수 있습니다. 이 부분은 크게 3 가지 측면으로 고찰됩니다.

1) Filter – 망파케트를 조종하는 기정려과방식입니다.
2) Nat – 망주소변환을 리용하여 새로운 련결을 진행합니다.
3) Mangle – 파케트변경의 특정한 형식을 사용합니다.
4) INPUT – 목표로되는 망 파케트에 적용됩니다.
5) OUTPUT – 국부적으로 생성되는 망파케트들에 적용됩니다.
6) FORWARD – 호스트를 통한 경로조종되는 망파케트에 적용됩니다.
7) PREROUTING – 파케트가 도착하면 변경합니다.
8) OUTPUT – 파케트가 수신되기 전에 국부적으로 생성된 망파케

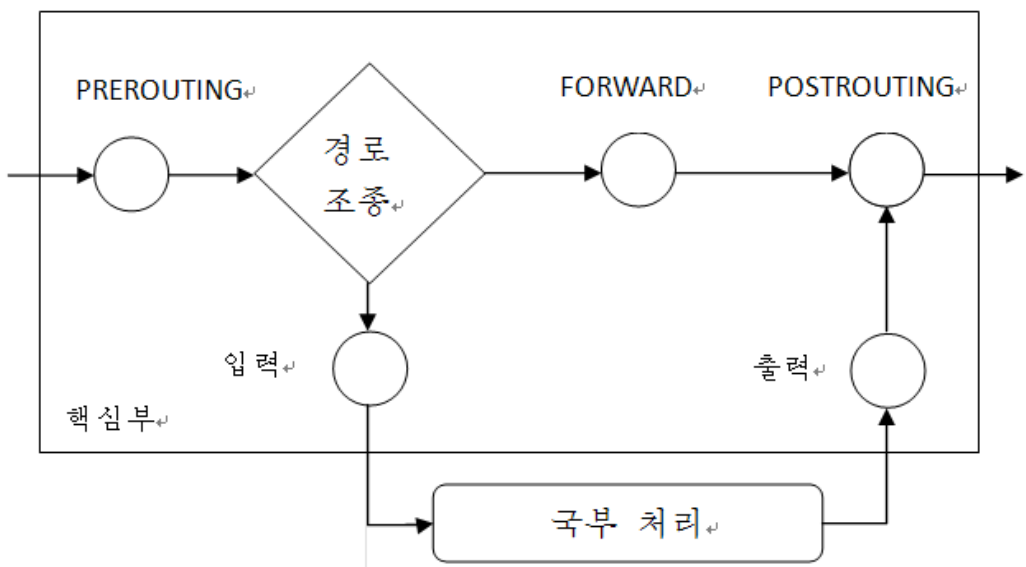
트들을 변경합니다.

9) POSTROUTING – 파케트가 수신되기 전에 망파케트들을 변경합니다.

Mangle 표를 위한 내장사슬은 다음과 같습니다.

1) INPUT – 호스트를 위한 목표파케트들을 변경합니다.
2) OUTPUT – 망파케트들이 수신되기전에 국부적으로 생성된 망파케트들을 변경합니다.
3) FORWARD – 호스트를 통하여 경로조종되는 망파케트들을 변경합니다.
4) PREROUTING – 망파케트들이 경로조종되기 전에 들어오는 망파케트들을 변경합니다.
5) POSTROUTING – 망파케트들이 수신되기 전에 변경합니다.

매 망파케트는 적어도 하나의 표에 의하여 봉사기체계에로 전송됩니다.

그러나 하나의 파케트에 사슬의 마지막결합전에 매 표에서 복합규칙이 적용될수도 있습니다. 이러한 규칙들의 구조와 목적은 다양하지만 보통 부분규약과 망봉사를 사용할때 주소의 설정이나 부분 IP 주소들에로 전송되는 하나의 파케트를 식별합니다. 다음의 그림은 iptables 부분체계에 의하여 파케트흐름이 어떻게 진행되는가를 보여줍니다.

2. IPTables 를 위한 지령항목

파케트려과를 위한 규칙들은 iptable 지령을 리용하여 생성됩니다. 지령들은 다음과 같습니다.

1) Packet Type – 려과되는 파케트의 형태를 특정화합니다.
2) Packet source/Destination – 파케트의 원천과 목표에 기초한 려과되는 파케트를 설정합니다.
3) Target – 우의 조건이 만족하는 파케트에 대하여 어떤 조작을 수행는가 하는것을 결정합니다.

1) IPTables 지령항목의 구조

구조는 다음과 같습니다.

iptables [-t <table-name>] <command> <chain-name> \ <parameter-1> <option-1> \ <parameter-n> <option-n>

<table-name> - 어느 표에서 규칙이 적용되였는가를 특정화합니다. 만일 생략되였으면 filter 표가 사용됩니다.

<command> - 규칙을 추가하거나 삭제하는것과 같은 동작들을 설정합니다.

<chain-name> - 편집, 생성, 삭제를 위한 사슬을 설정합니다.

<parameter>-<option> pairs – 규칙이 일치하는 하나의 파케트를 어떻게 처리하겠는가를 설정하는 파라메터와 련관항목들입니다.

Iptables 지령의 길이와 복잡성은 그의 목적에 따라 달라질수 있습니다.

실례로 사슬로부터 규칙을 삭제하는 지령은 매우 짧을수 있습니다.

Iptables –D <chain-name> <line-number>

이와는 대조적으로 여러가지 특정화된 파라메터들과 항목들을 리용하여 하나의 개별적인 부분망으로부터 파케트를 려과하는 규칙을 추가하는것은 매우 길수 있습니다. Iptables 지령의 구조는 유효한 규칙을

구성하는 파라메터들과 항목들을 요구한다는것이 중요합니다.

2) 지령항목

Iptables 지령에 하나의 지령항목이 허용됩니다. Help 지령을 제외하고 모든 지령들은 큰문자입니다.

지령들은 다음과 같습니다.

1) -A – 설정하려는 사슬의 끝에 규칙을 추가합니다.

2) -C – 사용자정의된 사슬에 개별적인 규칙을 추가하기전에 그것을 검사합니다. 이 지령은 추가적인 파라메터들과 항목들을 위한 iptables 규칙을 작성하는데 도움을 줍니다.

3) -D <integer> | <rule> - 설정한 번호에 해당한 개별적인 사슬에 있는 규칙을 삭제합니다. 이때 규칙설정은 반드시 존재하는 규칙에 하여야 합니다.

4) -E – 사용자정의된 사슬의 이름을 재설정합니다.

5) -F – 선택한 사슬을 맞춥니다. 이때 사슬에서 매 규칙을 효과적으로 삭제하는 사슬에 대하여 진행합니다. 만일 사슬이 설정되지 않으면 이 지령은 매 사슬로 부터 매 규칙을 지웁니다.

6) -h – 지령구조목록을 현시합니다.

7) -I [<integer>] – 사용자정의된 옹근수인수에 의하여 설정된 곳에 지정된 규칙을 삽입합니다. 만일 인수가 특정되여 있지 않으면 규칙은 사슬의 제일 우에 삽입됩니다.

8) -L – 그 지령뒤에 설정된 사슬에 있는 모든 규칙들을 목록화합니다. 기정 filter 표에 있는 모든 사슬들에 있는 모든 규칙들을 목록화하기 위하여 사슬 혹은 표를 특정화하지 않습니다. 다음의 실례는 개별적인 표에서 특정된 사슬에 있는 규칙들을 목록화하기 위하여 사용됩니다.

iptables-L<chain-name>-t<table-name>

9) -N – 사용자정의된 이름을 가진 새로운 사슬을 추가합니다. 사슬이름은 동일하여야 하며 그렇지 않은경우 오유통보문이 현시됩니다.

10) -P – 특정된 사슬을 위한 기정방책을 설정합니다.

그렇게 함으로써 규칙이 일치함이 없이 전반적인 사슬을 리행할때 ACCEPT, DROP 와 같은 설정된 목표에로 보내집니다.

11) -R – 특정화된 사슬에 있는 하나의 규칙을 교체합니다. 규칙의 번호는 사슬의 이름뒤에 설정되여야 합니다. 첫규칙은 1 번으로 합니다.

12) -X – 사용자정의된 사슬을 삭제합니다. 기정사슬은 삭제할수 없습니다.

13) -Z – 모든 사슬에 있는 바이트와 파케트 개수를 표에서 0 으로 설정합니다.

3) IPTables 파라메터 항목들

개별적인 사슬에서 규칙들의 추가, 삭제, 삽입, 교체 를 포함하는데 이것들은 파케트려파규칙을 구축하기 위한 여러가지 파라메터들을 요구합니다.

1) -c – 개별적인 규칙을 위한 계수를 재설정합니다. 이 파라메터는 PKTS 와 BYTES 항목을 접수합니다.

2) -d – 목적호스트이름, IP 주소, 규칙과 일치하는 파케트의 망을 설정합니다. 만일 하나의 망이 일치되면 다음의 IP 주소/망마스크형식이 지원됩니다.

N.N.N.N/M.M.M.M – N.N.N.N 은 IP 주소이고, M.M.M.M 은 망마스크입니다.

N.N.N.N/M – N.N.N.N 은 IP 주소이며, M 은 비트마스크입니다.

* -f – 토막화된 파케트들에만 이 규칙을 적용합니다.

* -I – eth0 혹은 ppp0 과 같은 들어오는 망대면부들을 설정합니다. Iptables 로 이 항목파라메터는 INPUT 와 FORWARD 사슬들로 리용될수 있는데 이때 filter 표와 PREROUTING 사슬이 리용됩니다.

* -j – 파케트가 개별적인 규칙과 맞으면 지정된 목표로 뛰여 넘습니다. 표준적인 목표들은 ACCEPT, DROP, QUEUE, RETURN 입니다.

4) IPTables 대응항목

망규약은 그 규약을 리용하는 파케트에 대응하도록 구성된 특정한

대응항목을 제공합니다. 그러나 그러한 규약은 먼저 iptables 지령으로 설정되여야 합니다. 실례로 –p <protocol-name> 항목이 설정됩니다. 규약이름 대신 규약 ID를 사용할수 있습니다.

iptables –A INPUT –p icmp –icmp-type any –j ACCEPT
iptables –A INPUT –p 5813 –icmp-type any –j ACCEPT

봉사정의는 /etc/services 화일에서 합니다.

5) 목표항목

파케트가 개별적인 규칙들과 대응되면 규칙은 서로다른 목표들에 어떠한 동작을 적용하겠는가를 결정하여야 합니다. 매 사슬은 기정목표를 가지고있습니다. 표준적인 목표들은 다음과 같습니다.

1) <user-defined-chain> - 표에 사용자정의된 사슬입니다. 이름은 유일하여야 하며 이 목표는 특정한 사슬로 파케트를 통과합니다.

2) ACCEPT – 목적지 혹은 다른 사슬을 통하여 파케트를 허용합니다.

3) DROP – 요구자에게 응답이 없이 파케트를 보냅니다. 파케트를 보내는 체계는 아무런 통지도 하지 않습니다.

4) QUEUE – 사용자공간응용프로그람에 의하여 조종되는 파케트를 대기렬조종합니다.

5) RETURN – 현재의 사슬에 있는 규칙들과 맞지 않는 파케트의 검사를 중지합니다.

6) LOG – 이 규칙과 대응되는 모든 파케트들을 리력합니다. /etc/syslog.conf 화일에 리력이 기록됩니다. 기정으로 /var/log/messages 화일에 위치합니다.

7) REJECT – 원격체계에 오유파케트를 전송하고 파케트를 차단합니다.